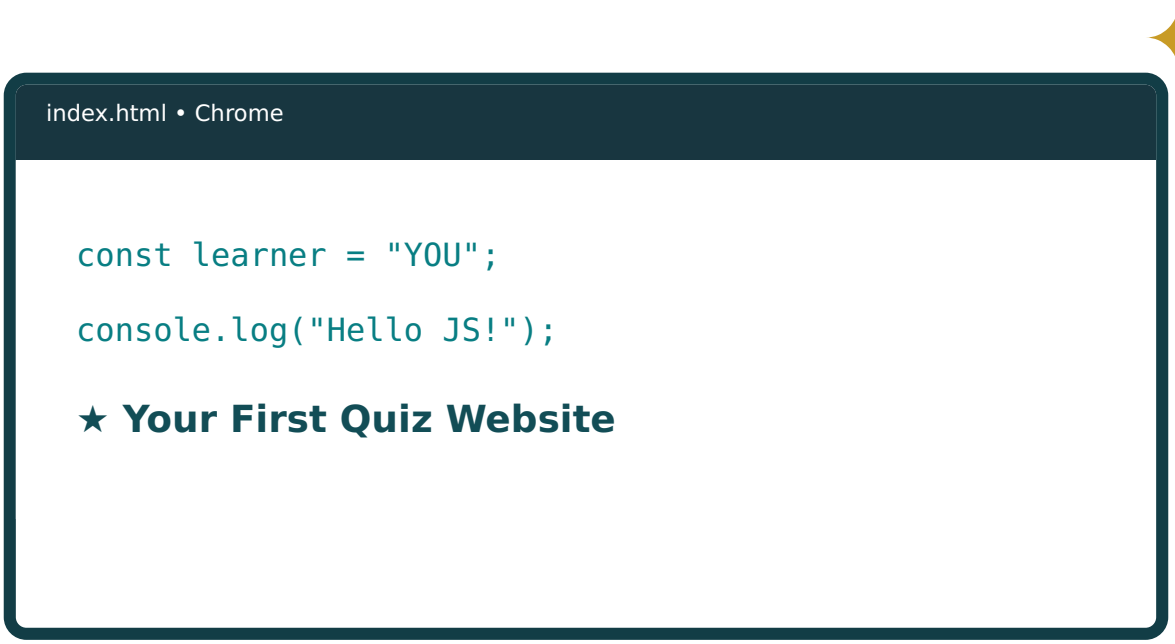
EGOTECHWORLD LEARNING SERIES | 2026

JAVASCRIPT • සිංහල • 2026

JavaScript for Beginners

මුල සිට ඉගෙනගනිමු

දරුවන්ට සහ ආරම්භකයන්ට සරල, ප්‍රායෝගික පාඩම්



<HTML> { CSS } JS( )

Little Star Quiz & Score Tracker

HTML • CSS • JAVASCRIPT • VS CODE • CHROME • TESTING

Pure white pages · dark code panels · practical exercises · verified source project

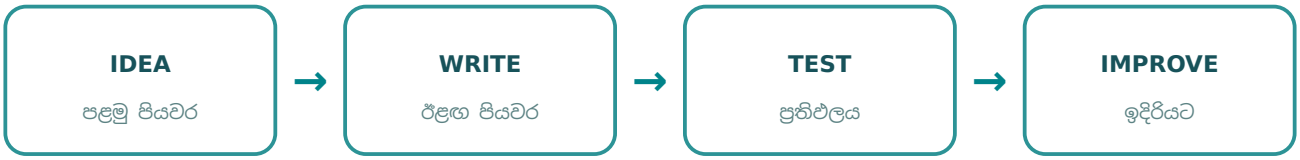
ගුරු සටහන

අපි හදන්නේ මොකක්ද?

මෙම පොතේ එක් එක් පාඩම ඉගෙනගෙන අවසානයේ ක්‍රියා කරන වෙබ් යෙදුමක් හදන්න.

- පළමුව JavaScript කියන්නේ කුමක්දැයි තේරුම් ගනිමු.
- දෙවනුව variables, functions, arrays, objects සහ conditions පුහුණු වෙමු.
- තුන්වනුව HTML + CSS සමඟ JavaScript එකතු කරමු.
- අවසානයේ ප්‍රශ්න 10ක් සහ ලකුණු ඇති Little Star Quiz website එකක් හදමු.

VISUAL LEARNING MAP



ACTIVITY

ඔබට කැමති පරිගණක game එකක button එකක් click කළාම සිදුවන දේ සටහන් කරන්න.

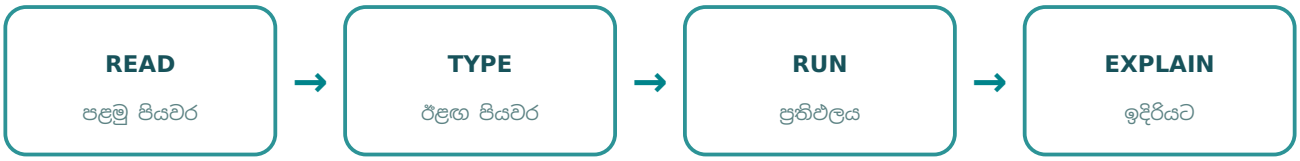
ගුරු සටහන

මෙම පොත භාවිතා කරන විදිහ

එක පිටුවක් කියවා එහි code එක ඔබේ පරිගණකයේ type කරන්න. Copy-paste කිරීමෙන් පමණක් නවතින්න එපා.

- 1. කියවන්න: සරල සිංහල පැහැදිලි කිරීම.
- 2. බලන්න: කළු VS Code panel එකේ code එක.
- 3. ධාවනය කරන්න: Chrome browser එකෙන් ප්‍රතිඵල බලන්න.
- 4. වෙනස් කරන්න: එක වචනයක් හෝ අංකයක් වෙනස් කර නැවත test කරන්න.
- 5. කියන්න: මිතුරෙකුට සිදු වූ දේ වචන දෙකකින් පැහැදිලි කරන්න.

VISUAL LEARNING MAP



ACTIVITY

ඔබේ පාඩම් සටහන සඳහා “JavaScript Journal” නමින් පිටුවක් අරඹන්න.

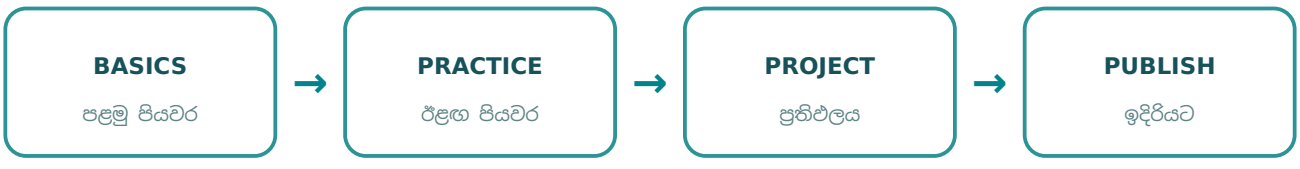
ගුරු සටහන

පොතේ අන්තර්ගතය

ඔබට අවශ්‍ය ඒකකය පහසුවෙන් සොයාගැනීමට මෙය භාවිතා කරන්න.

- UNIT 01 – වෙබ් පිටු සහ JavaScript මූලික කරුණු (පිටුව 8)
- UNIT 02 – Variables, numbers, decisions සහ loops (පිටුව 16)
- UNIT 03 – Functions, arrays, objects සහ JSON (පිටුව 37)
- UNIT 04 – Browser DOM, events සහ forms (පිටුව 53)
- UNIT 05 – Little Star Quiz project එක පියවරෙන් පියවර (පිටුව 69)
- UNIT 06 – QA, debugging, security සහ publishing (පිටුව 99)
- UNIT 07 – Modern JavaScript 2026, async සහ modules (පිටුව 111)
- අවසානය – MCQs, lab challenges, answer key සහ checklists

VISUAL LEARNING MAP



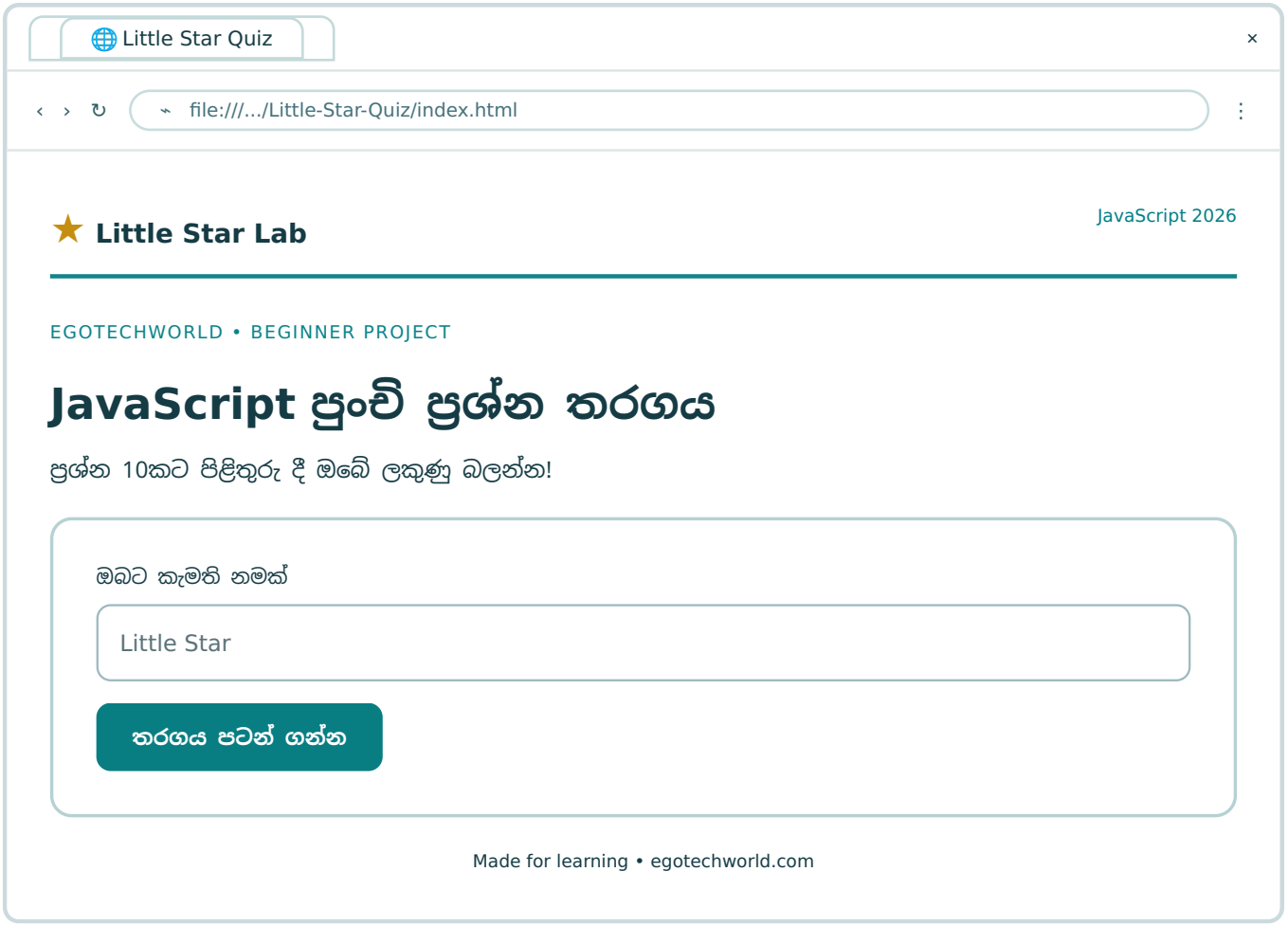
ACTIVITY

කැමති ඒකකයක් ලකුණු කර, ඉගෙනගත් පසු ✔ ලකුණ යොදන්න.

ගුරු සටහන

සැබෑ website එකක හැඩය

අපි නිර්මාණය කරන website එකේ welcome, question සහ result screen 3ක් තිබෙනවා.



පැහැදිලි කිරීම සඳහා නිර්මාණය කළ Chrome-style preview එකක්; browser version අනුව තිරය වෙනස් විය හැක.

- Welcome: කැමති nickname එකක් දී START කරන්න.
- Quiz: ප්‍රශ්නයකට options 4ක් අතරින් එකක් තෝරන්න.
- Result: 10න් ලබාගත් ලකුණු, feedback සහ correct answers බලන්න.
- Play again: නැවත අලුතින් අරඹන්න.
- Best score: browser localStorage එකේ ලකුණු පමණක් සුරකින්න.

ගුරු සටහන

අවශ්‍ය මෙවලම්

මෙම පාඩම් සඳහා ගෙවිය යුතු සේවාවක් හෝ database එකක් අවශ්‍ය නැහැ.

- Computer: Windows 10/11 හෝ Chrome සහ VS Code ධාවනය වන වෙනත් OS.
- Visual Studio Code: code files edit කරන්න.
- Google Chrome/Edge/Firefox: website එක බලන්න, DevTools භාවිතා කරන්න.
- Source ZIP: index.html, css සහ js files.
- Optional: Node.js – අග කොටසේ automated tests කිරීම සඳහා.

VISUAL LEARNING MAP



ACTIVITY

පරිගණකයේ text editor එකක් සහ browser එකක් විවෘත කර බලන්න.

ගුරු සටහන

සුරක්ෂිතව ඉගෙනගනිමු

මේක දරුවන්ගේ ඉගෙනීම සඳහා project එකක්. ඔබගේ හෝ වෙනත් කෙනෙකුගේ රහස් දත්ත කිසිදා code වලට නොයොදන්න.

- Passwords, bank details සහ සැබෑ student contact details source code එකට නොදමන්න.
- Quiz එකේ nickname එක අත්‍යවශ්‍ය නැහැ.
- Browser එකේ ඉතිරි කරන්නේ හොඳම ලකුණු අංකය පමණයි.
- Website එක GitHub Pages වෙත publish කළොත් code සහ files හැමෝටම දැකිය හැකියි.
- ගිණුම් නිර්මාණයට platform එකේ වයස් සීමා හා දෙමාපිය උපදෙස් අනුගමනය කරන්න.

VISUAL LEARNING MAP



ACTIVITY

පෞද්ගලික දත්ත සහ ඉගෙනීමේ sample data අතර වෙනස උදාහරණ 2කින් ලියන්න.

UNIT 01 • FIRST STEPS

Website එකක් කියන්නේ මොකක්ද?

වෙබ් පිටුවක් ගෙයක් වගේ හිතන්න. ඒ ගෙදර අස්ථි සැකිල්ල, පාට සහ ගමන් කරන දේවල් වෙන වෙනම භාර ගනී.

- HTML = පිටුවේ ව්‍යුහය: headings, paragraphs, buttons.
- CSS = වර්ණ, ඉඩ, අකුරු සහ සැලැස්ම.
- JavaScript = button එක click කළ විට සිදුවන ක්‍රියාව.
- මේ තුන එකට වැඩ කරන විට interactive website එකක් ලැබේ.

VISUAL LEARNING MAP



ACTIVITY

HTML නැත්නම් button එකක් කොහේද? CSS නැත්නම් එය කොහොම පෙනෙයිද?

UNIT 01 • FIRST STEPS

JavaScript කියන්නේ?

JavaScript කියන්නේ වෙබ් browser එක තුළ code ධාවනය කර පිටුවට හැසිරීමක් ලබාදෙන programming language එකක්.

- JavaScript සහ Java එකම language එකක් නොවෙයි.
- Browser එකේ JavaScript run වෙන්න වෙනම compiler එකක් මුල් පාඩමට අවශ්‍ය නැහැ.
- Input ගන්න, ගණන් කරන්න, button click වලට ප්‍රතිචාර දෙන්න පුළුවන්.
- Server මත Node.js මඟිනුත් JavaScript භාවිතා කළ හැකියි.



ACTIVITY

JavaScript භාවිතා විය හැකි app 3ක් ලියන්න.

UNIT 01 • FIRST STEPS

අපේ පළමු folder එක

VS Code විවෘත කර New Folder එකකට `hello-js` නම දෙන්න.

01 | New Folder → hello-js

02 | VS Code → File → Open Folder

03 | New File → index.html

04 | New File → script.js

05 | Save → Ctrl + S

ACTIVITY

folder path එක `hello-js/index.html` ලෙස සටහන් කරන්න.

UNIT 01 • FIRST STEPS

පළමු Hello World – Console

Chrome DevTools Console එකේ JavaScript message එකක් පෙන්වමු.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
console.log("Hello World!");
```

EXPECTED RESULT

Console panel: Hello World!

TRY IT YOURSELF

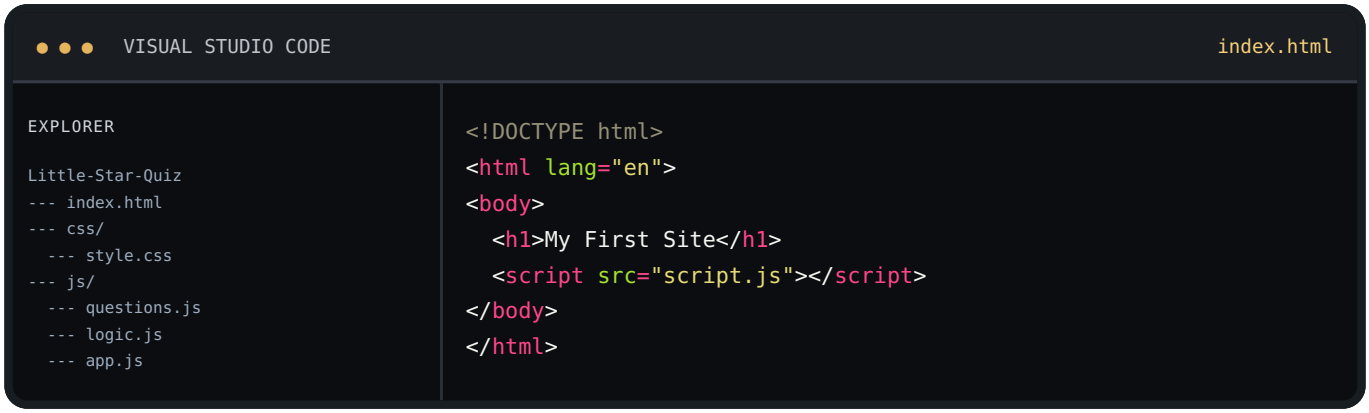
message එකට ඔබේ nickname එක යොදා නැවත run කරන්න.

මතක තබාගන්න: Chrome → F12 → Console; Warning messages ගැන කලබල නොවන්න.

UNIT 01 • FIRST STEPS

HTML සමඟ JavaScript

HTML file එකේ අවසානයේ script file එක එකතු කරමු.



EXPECTED RESULT

Chrome: My First Site; Console: script.js තුළ ලියා ඇති message

TRY IT YOURSELF

HTML file එක double-click කර browser එකේ බලන්න.

මතක තබාගන්න: Source files එකම folder එකේ තබන්න.

UNIT 01 • FIRST STEPS

Comments ලියන්නේ ඇයි?

Comment කියන්නේ code එක කියවන කෙනාට සටහනක්. Browser එක එය ක්‍රියාවට නංවන්නේ නැහැ.

VISUAL STUDIO CODE — script.js

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
// Show a simple greeting.
console.log("Good morning!");

/* This text explains
   the next calculation. */
console.log(2 + 3);
```

EXPECTED RESULT

Console: Good morning! සහ 5

TRY IT YOURSELF

ඔබේ code එකට English comment එකක් ලියන්න.

මතක තබාගන්න: Comments තුළ passwords තබන්න එපා.

UNIT 01 • FIRST STEPS

Code order සහ semicolons

Computer එක බොහෝ විට statements එකින් එක අනුපිළිවෙලට ක්‍රියා කරයි.

VISUAL STUDIO CODE — script.js

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
console.log("Step 1");
console.log("Step 2");
console.log("Step 3");
```

EXPECTED RESULT

Console: Step 1 → Step 2 → Step 3

TRY IT YOURSELF

Step 2 සහ Step 3 lines හුවමාරු කර බලන්න.

මතක තබාගන්න: JavaScript බොහෝ අවස්ථාවල semicolon නැතිව ක්‍රියා කළත් පාඩමට අනුකූලව භාවිතා කරමු.

UNIT 01 • FIRST STEPS

Error ආවොත් මොකද?

Programmers හැමෝටම errors එනවා. එය ඉගෙනගැනීමේ සාමාන්‍ය කොටසක්.

- SyntaxError: quotes, brackets හෝ spelling එක අඩු විය හැක.
- ReferenceError: නම වැරදි ලෙස ලියා තිබිය හැක.
- Console එකේ error එකේ file name සහ line number බලන්න.
- වෙනස්කම එකක් පමණක් කර නැවත run කරන්න.



ACTIVITY

මෙහි අඩු `)` ලකුණ සොයා නිවැරදි කරන්න.

UNIT 02 • THINKING WITH CODE

Variable එකක් කියන්නේ පෙට්ටියක්

නමක් යොදා අගයක් තබන කුඩා පෙට්ටියක් වගේ variable එකක් හිතන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
let points = 10;
console.log(points);
points = 15;
console.log(points);
```

EXPECTED RESULT

10 සහ 15

TRY IT YOURSELF

points = 20 කර console එකෙන් බලන්න.

මතක තබාගන්න: `let` භාවිතා කළ විට අගය පසුව වෙනස් කළ හැකියි.

UNIT 02 • THINKING WITH CODE

const – වෙනස් නොවන label

නැවත වෙනස් නොකළ යුතු variable binding එකකට `const` භාවිතා කරමු.

VISUAL STUDIO CODE

script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const quizTitle = "Little Star Quiz";
console.log(quizTitle);
// quizTitle = "Other"; // Error
```

EXPECTED RESULT

Little Star Quiz

TRY IT YOURSELF

`const` සහ `let` වෙනස ඔබේ වචනවලින් කියන්න.

මතක තබාගන්න: Object එකක properties වෙනස් කළ හැකි වුවත් const binding එක නැවත assign කළ නොහැක.

UNIT 02 • THINKING WITH CODE

String – අකුරු තබන පෙට්ටිය

අකුරු හෝ වචන double quotes හෝ single quotes තුළ ලියන්න.



EXPECTED RESULT

Hello Nila

TRY IT YOURSELF

name value එකට වෙනත් nickname එකක් යොදන්න.

මතක තබාගන්න: String සහ number එක අතර වෙනස හඳුනාගන්න.

UNIT 02 • THINKING WITH CODE

Number – ගණිතය කරන අගය

ගණිත ක්‍රියා සඳහා number type එක භාවිතා කරමු.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
let apples = 3;
let oranges = 4;
console.log(apples + oranges);
console.log(oranges - apples);
```

EXPECTED RESULT

7 සහ 1

TRY IT YOURSELF

apples 5 කර අලුත් ප්‍රතිඵලය ලියන්න.

මතක තබාගන්න: Number එක quotes තුළ දැමුවොත් එය string එකක් විය හැක.

UNIT 02 • THINKING WITH CODE

+, -, *, / operators

JavaScript වලින් එකතු කිරීම්, අඩු කිරීම් සහ බෙදීම් කරමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
console.log(6 + 2);
console.log(6 - 2);
console.log(6 * 2);
console.log(6 / 2);
```

EXPECTED RESULT

8, 4, 12, 3

TRY IT YOURSELF

7 * 3 සහ 12 / 4 ගණන් කරන්න.

මතක තබාගන්න: Math operators සාමාන්‍ය ගණිත අනුපිළිවෙල අනුගමනය කරයි.

UNIT 02 • THINKING WITH CODE

String + number වෙනස

Input text එකක් number ලෙස පරිවර්තනය නොකළොත් වැරදි එකතුවක් ලැබිය හැකියි.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
console.log("4" + 2);
console.log(Number("4") + 2);
```

EXPECTED RESULT

42 සහ 6

TRY IT YOURSELF

"10" + 5 හි ප්‍රතිඵලය අනුමාන කරන්න.

මතක තබාගන්න: Form input.value බොහෝවිට string එකක් බව මතක තබාගන්න.

UNIT 02 • THINKING WITH CODE

Boolean – ඔව් / නැහැ

Boolean අගයක් true හෝ false ලෙස දෙකෙන් එකක් වේ.

VISUAL STUDIO CODE script.js

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const isReady = true;
const isFinished = false;
console.log(isReady);
console.log(isFinished);
```

EXPECTED RESULT

true සහ false

TRY IT YOURSELF

isLoggedIn නමින් false variable එකක් හදන්න.

මතක තබාගන්න: true සහ false quotes තුළ දැමුවොත් boolean නොව string වේ.

UNIT 02 • THINKING WITH CODE

typeof – අගය මොන වර්ගයේද?

Value එකේ type එක console එකෙන් පරීක්ෂා කරමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
console.log(typeof "Hello");
console.log(typeof 42);
console.log(typeof true);
```

EXPECTED RESULT

string, number, boolean

TRY IT YOURSELF

typeof 3.14 හි output එක අනුමාන කරන්න.

මතක තබාගන්න: beginner project එකේ type ගැන අවබෝධය errors අඩු කරයි.

UNIT 02 • THINKING WITH CODE

Template literal සමඟ greeting

Backticks භාවිතයෙන් string එකක් තුළ variable එක සරලව යොදන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const name = "Mina";
const stars = 5;
console.log(`Hi ${name}! Stars: ${stars}`);
```

EXPECTED RESULT

Hi Mina! Stars: 5

TRY IT YOURSELF

ඔබේ nickname එක දමා greeting එකක් සකස් කරන්න.

මතක තබාගන්න: Backtick (`) සහ single quote (') එක සමාන නොවෙයි.

UNIT 02 • THINKING WITH CODE

Comparison – සසඳන්න

Values දෙකක් සසඳා true හෝ false ලබාගත හැකියි.

VISUAL STUDIO CODE script.js

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
console.log(5 > 3);
console.log(5 < 3);
console.log(5 === 5);
console.log(5 !== 3);
```

EXPECTED RESULT

true, false, true, true

TRY IT YOURSELF

8 >= 8 අගය අනුමාන කරන්න.

මතක තබාගන්න: === සමාන value සහ type බලන නිසා beginnersට හොඳ තේරීමක්.

UNIT 02 • THINKING WITH CODE

if – තීරණයක් ගන්න

Condition එක true නම් එහි ඇතුළත code block එක ධාවනය වේ.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const age = 12;
if (age >= 10) {
  console.log("You can try the quiz!");
}
```

EXPECTED RESULT

You can try the quiz!

TRY IT YOURSELF

age = 9 කර නැවත බලන්න.

මතක තබාගන්න: මෙය learning example එකක්; සැබෑ සේවා වයස් නීති වෙනම තිබිය හැක.

UNIT 02 • THINKING WITH CODE

if / else – මාර්ග දෙකක්

කොන්දේසිය නිවැරදි නැතිනම් else කොටස ක්‍රියා කරයි.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const score = 6;
if (score >= 5) {
  console.log("Great work!");
} else {
  console.log("Try again!");
}
```

EXPECTED RESULT

Great work!

TRY IT YOURSELF

score = 3 කර output එක ලියන්න.

මතක තබාගන්න: Conditions පමණක් භාවිතා කර පුද්ගලයෙකු ගැන අසාධාරණ තීරණ නොගන්න.

UNIT 02 • THINKING WITH CODE

else if – අවස්ථා තුනක්

Condition එකකට වඩා වැඩි තීරණ තිබුණොත් else if භාවිතා කළ හැකියි.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const marks = 74;
if (marks >= 80) {
  console.log("A");
} else if (marks >= 60) {
  console.log("B");
} else {
  console.log("Practice");
}
```

EXPECTED RESULT

B

TRY IT YOURSELF

marks = 92, marks = 45 වෙනස් කර බලන්න.

මතක තබාගන්න: if/else-if අනුපිළිවෙල වැදගත්.

UNIT 02 • THINKING WITH CODE

&& – දෙකම හරිද?

දෙකම true වීම අවශ්‍ය විට AND (`&&`) භාවිතා කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const hasPencil = true;
const hasBook = true;
if (hasPencil && hasBook) {
  console.log("Ready to learn!");
}
```

EXPECTED RESULT

Ready to learn!

TRY IT YOURSELF

hasBook = false කළ විට මොකද වෙන්නේ?

මතක තබාගන්න: &&, ||, ! තුන වෙන වෙනම මතක තබාගන්න.

UNIT 02 • THINKING WITH CODE

|| - එකක් හරිද?

Condition දෙකෙන් එකක් true නම් OR (`||`) true වේ.

VISUAL STUDIO CODE

script.js

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const likesCats = false;
const likesDogs = true;
if (likesCats || likesDogs) {
  console.log("Loves animals!");
}
```

EXPECTED RESULT

Loves animals!

TRY IT YOURSELF

දෙකම false කළ විට result එක අනුමාන කරන්න.

මතක තබාගන්න: || කියන්නේ OR; `!` කියන්නේ NOT.

UNIT 02 • THINKING WITH CODE

for loop – නැවත නැවත

එකම code එක කිහිප වරක් ධාවනය කිරීමට loop එකක් භාවිතා කරමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
for (let i = 1; i <= 5; i++) {
  console.log(`Star ${i}`);
}
```

EXPECTED RESULT

Star 1, Star 2, Star 3, Star 4, Star 5

TRY IT YOURSELF

5 වෙනුවට 3 යොදන්න.

මතක තබාගන්න: Condition එක අවසන් වන ආකාරය සැලසුම් නොකළොත් loop එක නතර නොවිය හැක.

UNIT 02 • THINKING WITH CODE

while loop – තවම අවශ්‍යද?

Condition එක true වන තෙක් while loop එක ඉදිරියට යයි.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
let count = 1;
while (count <= 3) {
  console.log(count);
  count += 1;
}
```

EXPECTED RESULT

1, 2, 3

TRY IT YOURSELF

count += 1 line එක ඉවත් නොකරන්න; එය loop එක අවසන් කරයි.

මතක තබාගන්න: while loop එකෙන් infinite loop එකක් නොසෑදෙන්න බලන්න.

UNIT 02 • THINKING WITH CODE

Counter එකක් හදමු

Score tracker එකේ score වැඩි කිරීමට variable එකක් භාවිතා කරමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
let score = 0;
score += 1;
score += 1;
console.log(score);
```

EXPECTED RESULT

2

TRY IT YOURSELF

score -= 1 line එකක් එකතු කර බලන්න.

මතක තබාගන්න: score += 1 කියන්නේ score = score + 1 වගේයි.

UNIT 02 • THINKING WITH CODE

Math.round – සරල ප්‍රතිශතය

ප්‍රශ්න 10කින් ලබාගත් ලකුණු percentage ලෙස පෙන්වමු.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const score = 7;
const total = 10;
const percent = Math.round(score / total * 100);
console.log(percent + "%");
```

EXPECTED RESULT

70%

TRY IT YOURSELF

9/10 සහ 3/5 percentage ගණනය කරන්න.

මතක තබාගන්න: total = 0 විය හැකි තැන්වල මුලින්ම check කරන්න.

UNIT 02 • THINKING WITH CODE

අංකයක් හරිද?

Number.isFinite වලින් NaN හෝ Infinity වැනි අගයන් පරීක්ෂා කළ හැක.

VISUAL STUDIO CODE — script.js

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const a = Number("42");
const b = Number("hello");
console.log(Number.isFinite(a));
console.log(Number.isFinite(b));
```

EXPECTED RESULT

true සහ false

TRY IT YOURSELF

Number("") හි result එක console එකෙන් බලන්න.

මතක තබාගන්න: Input validation දරුවන්ටත් සරලව ඉගැන්විය හැකි වැදගත් QA පුරුද්දකි.

UNIT 02 • QUICK CHECK

Variables සහ conditions – MCQ 01

පැන්සලක් හෝ පිටුපස සටහන් පොතක් භාවිතා කර පිළිතුරු ලියන්න.

1. වෙනස් කළ හැකි variable binding එකක් සඳහා?

A. let • B. close • C. HTML

2. `const` එකක් නැවත assign කළ හැකිද?

A. ඔව් • B. නැහැ

3. "5" + 2 හි ප්‍රතිඵලය?

A. 7 • B. 52 • C. 10

4. `===` භාවිතා කරන්නේ?

A. සසඳන්න • B. print කරන්න • C. database delete කරන්න

CHALLENGE

පිළිතුරු අග කොටසේ Answer Key එකේ බලන්න.

UNIT 03 • REUSABLE IDEAS

Function – නැවත භාවිතා කරන මෙවලම

එකම code කොටස නැවත නැවත භාවිතයට function එකක් හදමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function sayHello() {
  console.log("Hello, friend!");
}
sayHello();
sayHello();
```

EXPECTED RESULT

Hello, friend! දෙවරක්

TRY IT YOURSELF

sayBye() නමින් function එකක් නිර්මාණය කරන්න.

මතක තබාගන්න: Function එක define කිරීම සහ call කිරීම දෙක වෙනස් දේවල්.

UNIT 03 • REUSABLE IDEAS

Parameters – ඇතුළට දෙන දත්ත

Function එකට පිටතින් නමක් හෝ අගයක් යවන්න පුළුවන්.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function greet(name) {
  console.log(`Hello ${name}!`);
}
greet("Asha");
greet("Nimal");
```

EXPECTED RESULT

Hello Asha! / Hello Nimal!

TRY IT YOURSELF

ඔබේ nickname එක argument එකක් ලෙස යොදන්න.

මතක තබාගන්න: name යනු parameter; "Asha" යනු argument.

UNIT 03 • REUSABLE IDEAS

return – පිළිතුර ආපසු දෙන්න

Function එකක් ගණනය කර result එක ආපසු දෙන්න return යොදාගනී.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function add(a, b) {
  return a + b;
}
const total = add(3, 4);
console.log(total);
```

EXPECTED RESULT

7

TRY IT YOURSELF

multiply(a, b) function එකක් හදන්න.

මතක තබාගන්න: return එකෙන් පසු එම function call එකේ ඉතිරි statements ක්‍රියා නොකරයි.

UNIT 03 • REUSABLE IDEAS

Arrow function සරල හැඳින්වීම

අලුත් JavaScript code වල arrow syntax බහුලව පෙනේ.

VISUAL STUDIO CODE script.js

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const square = (n) => n * n;
console.log(square(5));
```

EXPECTED RESULT

25

TRY IT YOURSELF

double = (n) => n * 2 ලෙස හදන්න.

මතක තබාගන්න: Arrow functions සහ සාමාන්‍ය functions අතර `this` හැසිරීම වැනි වෙනස්කම් ඇත; එය පසුව ඉගෙනගන්න.

UNIT 03 • REUSABLE IDEAS

Scope – variable එකේ සීමාව

Block එකක් ඇතුළේ ඇති let variable එකක් එම block එකට පමණක් සීමා විය හැක.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
if (true) {
  let secretNumber = 12;
  console.log(secretNumber);
}
// console.log(secretNumber); // Error
```

EXPECTED RESULT

12; අවසාන line එක uncomment කළොත් error

TRY IT YOURSELF

secretNumber ඉදිරියට පෙන්වීමට code පිහිටීම වෙනස් කරන්න.

මතක තබාගන්න: scope ඉගෙනගත් විට නම ගැටලු අඩු වේ.

UNIT 03 • REUSABLE IDEAS

Array – දත්ත ලැයිස්තුව

බොහෝ අගයන් එකට තබාගන්න array භාවිතා කරමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const fruits = ["Mango", "Apple", "Banana"];
console.log(fruits[0]);
console.log(fruits.length);
```

EXPECTED RESULT

Mango සහ 3

TRY IT YOURSELF

fruits[1] හි පිළිතුර ලියන්න.

මතක තබාගන්න: Array index පටන්ගන්නේ 0 න් බව මතක තබාගන්න.

UNIT 03 • REUSABLE IDEAS

Array push සහ pop

අලුත් item එකක් එකතු කිරීමට push; අවසාන item ඉවත් කිරීමට pop.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const tasks = ["Read", "Code"];
tasks.push("Test");
console.log(tasks);
tasks.pop();
console.log(tasks);
```

EXPECTED RESULT

[Read, Code, Test] → [Read, Code]

TRY IT YOURSELF

tasks.unshift("Plan") කර බලන්න.

මතක තබාගන්න: අපේ quiz ප්‍රශ්න data එකත් array එකකින් ගන්නවා.

UNIT 03 • REUSABLE IDEAS

for...of සමඟ array කියවීම

Array එකේ item එකින් එක loop එකෙන් බලන්න.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const colors = ["Red", "Green", "Blue"];
for (const color of colors) {
  console.log(color);
}
```

EXPECTED RESULT

Red, Green, Blue

TRY IT YOURSELF

වර්ණ 4ක් දමා for...of run කරන්න.

මතක තබාගන්න: Array එකේ items ගණන දැන සිටිය යුතු නැහැ.

UNIT 03 • REUSABLE IDEAS

Object – එක් දේකට ගුණාංග

පුද්ගලයෙකු හෝ භාණ්ඩයක් ගැන ගුණාංග කිහිපයක් එක object එකකට දමමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const student = {
  nickname: "Star",
  points: 8
};
console.log(student.nickname);
```

EXPECTED RESULT

Star

TRY IT YOURSELF

student.points කියවා console එකේ පෙන්වන්න.

මතක තබාගන්න: Object එකක් තුළ key:value යුගල ඇත.

UNIT 03 • REUSABLE IDEAS

Array තුළ Objects

Quiz project එකේ ප්‍රශ්නයක් සහ options එකට එකතු කිරීමට array of objects භාවිතා කරමු.



EXPECTED RESULT

2 + 2?

TRY IT YOURSELF

තුන්වැනි question එකක් එකතු කරන්න.

මතක තබාගන්න: Real project එකේ එක් ප්‍රශ්නයකට options 4ක් ද ඇත.

UNIT 03 • REUSABLE IDEAS

String methods

අකුරු පිරිසිදු කිරීමට, කුඩා අකුරු කරන්න, කොටස් සොයන්න methods යොදාගත හැක.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const name = "  Little STAR  ";
console.log(name.trim());
console.log(name.toLowerCase());
console.log(name.includes("STAR"));
```

EXPECTED RESULT

Little STAR / " little star " / true

TRY IT YOURSELF

String එකේ .length කියන්නේ මොකක්ද?

මතක තබාගන්න: trim() මඟින් දෙපස අමතර spaces ඉවත් කරයි.

UNIT 03 • REUSABLE IDEAS

Array map – අලුත් ලැයිස්තුව

map මඟින් එක් එක් අගයට ක්‍රියාවක් කර අලුත් array එකක් ගන්නවා.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const scores = [2, 4, 6];
const doubled = scores.map(n => n * 2);
console.log(doubled);
```

EXPECTED RESULT

[4, 8, 12]

TRY IT YOURSELF

නව values වලට 1 එකතු කර බලන්න.

මතක තබාගන්න: map මුල් array එක වෙනස් නොකර අලුත් array එකක් ලබාදේ.

UNIT 03 • REUSABLE IDEAS

Array filter – තෝරාගැනීම

Filter condition එක pass කරන values පමණක් ගන්නවා.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const numbers = [1, 2, 3, 4, 5];
const big = numbers.filter(n => n > 3);
console.log(big);
```

EXPECTED RESULT

[4, 5]

TRY IT YOURSELF

even numbers ගන්න `n % 2 === 0` යොදන්න.

මතක තබාගන්න: filter result එකෙන් සියල්ල හෝ කිසිවක් හෝ ලැබිය හැකියි.

UNIT 03 • REUSABLE IDEAS

JSON කියන්නේ මොකක්ද?

JSON කියන්නේ data එක text ලෙස හුවමාරු කිරීමට භාවිතා කරන සම්මත රටාවක්.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const student = { name: "Star", score: 10 };
const json = JSON.stringify(student);
console.log(json);
console.log(JSON.parse(json).score);
```

EXPECTED RESULT

JSON string එකක් සහ 10

TRY IT YOURSELF

Object එකට className property එකක් එකතු කරන්න.

මතක තබාගන්න: JSON keys double quotes සමඟ text ලෙස ගොඩනගනවා.

UNIT 03 • REUSABLE IDEAS

Pure function – පරීක්ෂා කරන්න ලේසි

Browser එකේ screen වෙනස් නොකර result එකක් ලබාදෙන function එකක් test කිරීම පහසුයි.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
function percent(score, total) {
  if (total <= 0) return 0;
  return Math.round(score / total * 100);
}
console.log(percent(3, 5));
```

EXPECTED RESULT

60

TRY IT YOURSELF

percent(0, 0) safe result එක ලියන්න.

මතක තබාගන්න: මෙවැනි functions tests වලට වඩාත් හොඳයි.

UNIT 03 • QUICK CHECK

Functions සහ arrays – MCQ 02

පැන්සලක් හෝ පිටුපස සටහන් පොතක් භාවිතා කර පිළිතුරු ලියන්න.

1. Array එකක පළමු index එක?

A. 1 • B. 0 • C. 10

2. Function එකෙන් පිළිතුරක් දෙන්නේ?

A. return • B. color • C. width

3. Array එකේ අගට item දමන්නේ?

A. push • B. if • C. const

4. Object එකක දත්ත ලියන්නේ?

A. key:value • B. <p> • C. file.exe

CHALLENGE

නොදන්නා වචනයක් තිබුණොත් lesson එකේ සටහන් වෙත ආපසු යන්න.

UNIT 04 • BROWSER MAGIC

DOM කියන්නේ?

HTML page එක browser එක ගසක් වගේ දත්ත ව්‍යුහයක් ලෙස පරිවර්තනය කරනවා. JavaScript එකෙන් ඒ elements සොයා වෙනස් කළ හැක.

- HTML elements = ගසේ අතු වගේ nodes.
- document කියන්නේ browser එකේ page object එක.
- getElementById සහ querySelector මඟින් element සොයන්න.
- textContent මඟින් text වෙනස් කරන්න.

VISUAL LEARNING MAP



ACTIVITY

h1 element එකක් සහ button element එකක් DOM tree එකේ චිත්‍රයක් ලෙස අඳින්න.

UNIT 04 • BROWSER MAGIC

getElementById()

HTML element එකකට id එකක් දුන් විට JavaScript එකෙන් එය සොයාගන්න පුළුවන්.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```html
<h1 id="title">Hello</h1>
<script>
  const heading = document.getElementById("title");
  heading.textContent = "Hello, Star!";
</script>
```

EXPECTED RESULT

Browser heading: Hello, Star!

TRY IT YOURSELF

id එක title වෙනුවට greeting කරන්න; JS එකත් ගළපන්න.

මතක තබාගන්න: id එක document එකේ අනන්‍ය විය යුතුයි.

UNIT 04 • BROWSER MAGIC

textContent භාවිතා කරන්න

User-entered text එකක් පෙන්වීමට textContent භාවිතා කිරීමෙන් HTML code ලෙස නොසලකයි.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const output = document.getElementById("output");
output.textContent = "Welcome!";
```

EXPECTED RESULT

output element එකේ Welcome! දිස්වේ.

TRY IT YOURSELF

textContent සහ innerHTML අතර වෙනස සටහන් කරන්න.

මතක තබාගන්න: සාමාන්‍ය text පෙන්වීමට textContent ආරක්ෂිත තේරීමකි.

UNIT 04 • BROWSER MAGIC

querySelector()

CSS selector එකක් භාවිතා කර පළමු match වන element එක ගන්නවා.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const firstButton = document.querySelector("button");
firstButton.textContent = "Start Quiz";
```

EXPECTED RESULT

පළමු button එකේ text වෙනස් වේ.

TRY IT YOURSELF

class selector `.start-btn` එක්ක අත්හදා බලන්න.

මතක තබාගන්න: Selector එකට ගැළපෙන element නැත්නම් null ලැබේ.

UNIT 04 • BROWSER MAGIC

button click event එකක්

Button එක click කරන විට function එකක් run කිරීමට addEventListener යොදමු.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```html
<button id="hello">Click me</button>
<script>
  document.getElementById("hello")
    .addEventListener("click", () => {
      alert("Hi!");
    });
</script>
```

EXPECTED RESULT

Button click → alert Hi!

TRY IT YOURSELF

alert වෙනුවට textContent update එකක් භාවිතා කරන්න.

මතක තබාගන්න: HTML inline onclick වෙනුවට addEventListener භාවිතා කිරීම හොඳ පුරුද්දකි.

UNIT 04 • BROWSER MAGIC

Input field එකෙන් text ගන්න

Input එකේ value property එක කියවා greeting එකක් පෙන්වමු.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```html
<input id="nick" placeholder="Nickname">
<button id="go">Greet</button>
<p id="out"></p>
<script>
  const go = document.getElementById("go");
  const nick = document.getElementById("nick");
  const out = document.getElementById("out");
  go.addEventListener("click", () => {
    out.textContent = `Hi ${nick.value.trim()}`;
  });
</script>
```

EXPECTED RESULT

Nickname → Hi Nila

TRY IT YOURSELF

හිස් input එකක් ආවොත් default name එකක් පෙන්වන්න.

මතක තබාගන්න: DOM elements getElementById මඟින් පැහැදිලිව සොයාගෙන භාවිතා කරමු.

UNIT 04 • BROWSER MAGIC

Input validation

User type කරන input එක හැමවිටම නිවැරදි කියා නොහිතන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const input = document.getElementById("age");
const age = Number(input.value);
if (!Number.isFinite(age) || age < 0) {
  console.log("Enter a valid age");
}
```

EXPECTED RESULT

වැරදි input එකක් සඳහා error message

TRY IT YOURSELF

හිස් string එකක් Number("") කළ විට 0 ලැබෙන බව බලන්න.

මතක තබාගන්න: අවශ්‍ය නම් empty input වෙනම check කරන්න.

UNIT 04 • BROWSER MAGIC

classList – පාටක් වෙනස් කරන්න

CSS class එකක් add කර element එකක හැඩය වෙනස් කළ හැක.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const card = document.getElementById("card");
card.classList.add("active");
card.classList.remove("hidden");
```

EXPECTED RESULT

card element එකට active class එක ලැබේ.

TRY IT YOURSELF

active වෙනුවට correct class එකක් නිර්මාණය කරන්න.

මතක තබාගන්න: Design CSS එකේ; behavior JavaScript එකේ තබන එක පහසුයි.

UNIT 04 • BROWSER MAGIC

hidden – screen 3ක් පාලනය

අපේ quiz එකේ welcome, quiz, result කොටස් මාරු කිරීමට hidden property භාවිතා කරමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
welcome.hidden = true;
quiz.hidden = false;
result.hidden = true;
```

EXPECTED RESULT

Welcome නොපෙනී Quiz screen එක පෙනේ.

TRY IT YOURSELF

Result screen පෙන්වන code එක ලියන්න.

මතක තබාගන්න: Project එකේ show(sectionId) function එක මඟින් මේ ත්‍රිත්වය පාලනය වේ.

UNIT 04 • BROWSER MAGIC

createElement() – අලුත් button

JS මඟින් programmatically HTML element එකක් හදන්න පුළුවන්.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const button = document.createElement("button");
button.textContent = "Answer A";
document.body.appendChild(button);
```

EXPECTED RESULT

Browser එකට Answer A button එක එකතු වේ.

TRY IT YOURSELF

button.textContent value එක වෙනස් කරන්න.

මතක තබාගන්න: User text එක HTML ලෙස assemble නොකර createElement භාවිතා කරන්න.

UNIT 04 • BROWSER MAGIC

forEach() සමඟ answer buttons

Options හතර වෙන වෙනම type නොකර array එකෙන් buttons හදමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const options = ["A", "B", "C", "D"];
options.forEach((text) => {
  const b = document.createElement("button");
  b.textContent = text;
  answers.appendChild(b);
});
```

EXPECTED RESULT

A, B, C, D buttons හතරක්

TRY IT YOURSELF

ඒ එක් button එකකට click event එකක් දමන්න.

මතක තබාගන්න: `answers` යනු කලින් සොයාගත් HTML container element එකක් විය යුතුයි.

UNIT 04 • BROWSER MAGIC

preventDefault() – form submit

Form එක submit වූ විට browser එක reload නොවීමට preventDefault යොදාගත හැක.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
form.addEventListener("submit", (event) => {
  event.preventDefault();
  console.log("We will handle this!");
});
```

EXPECTED RESULT

Submit → Console message; page reload නොවේ.

TRY IT YOURSELF

submit වෙනුවට click event එකක් ලියන්න.

මතක තබාගන්න: Form event එකක් භාවිතා කළාම Enter key එකෙනුත් submit විය හැක.

UNIT 04 • BROWSER MAGIC

Keyboard භාවිතා කරන අයත් සිතන්න

Website එක mouse එකෙන් පමණක් නොව keyboard එකෙනුත් භාවිතා කළ හැකි විය යුතුයි.

VISUAL STUDIO CODE — script.js

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const btn = document.createElement("button");
btn.type = "button";
btn.textContent = "Start";
btn.addEventListener("click", () => alert("Started!"));
document.body.appendChild(btn);
```

EXPECTED RESULT

Tab → focus; Enter → click

TRY IT YOURSELF

Tab සහ Enter භාවිතා කර website එක පරීක්ෂා කරන්න.

මතක තබාගන්න: Real <button> element එකක keyboard behavior browser එක සපයයි.

UNIT 04 • BROWSER MAGIC

LocalStorage – මතක තබාගැනීම

Browser එකේ කුඩා data කැබැල්ලක් සුරැකීමට localStorage භාවිතා කරන්න පුළුවන්.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
localStorage.setItem("best", "8");
const best = localStorage.getItem("best");
console.log(best);
```

EXPECTED RESULT

"8" (string එකක්)

TRY IT YOURSELF

localStorage.removeItem("best") කර බලන්න.

මතක තබාගන්න: LocalStorage රහස් හෝ passwords සඳහා සුදුසු නැහැ.

UNIT 04 • BROWSER MAGIC

try / catch – දෝෂයකදීත් වැඩ කරයි

Storage blocked වුණත් quiz app එක නතර නොවෙන්න try/catch යොදමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
let best = 0;
try {
  best = Number(localStorage.getItem("best")) || 0;
} catch (error) {
  best = 0;
}
console.log(best);
```

EXPECTED RESULT

Saved score හෝ 0

TRY IT YOURSELF

මේ code එක නොමැතිව storage error එකක් ආවොත් සිදුවිය හැකි දේ ලියන්න.

මතක තබාගන්න: Error catch කළාට පසු user ට වැරදි දත්ත පෙන්වන්න එපා.

UNIT 04 • QUICK CHECK

DOM සහ Events – MCQ 03

පැන්සලක් හෝ පිටුපස සටහන් පොතක් භාවිතා කර පිළිතුරු ලියන්න.

1. HTML element එකක් සොයාගන්න?

A. getElementById • B. Math.ceil • C. console.clear

2. Text පමණක් පෙන්වීමට?

A. innerHTML • B. textContent

3. Button click එකට?

A. addEventListener • B. setTimeout • C. JSON.parse

4. localStorage තුළ passwords තබන එක හොඳද?

A. ඔව් • B. නැහැ

CHALLENGE

ඔබේ browser එකේ F12 Console එක විවෘත කිරීමත් පුහුණු වෙන්න.

UNIT 05 • LITTLE STAR QUIZ

Project එකේ සැලැස්ම

අපි එකම දේ පියවරෙන් පියවර හදමු. ඉක්මන් කරන්න අවශ්‍ය නැහැ.

01 | 1. HTML structure ලියන්න

02 | 2. CSS design එකතු කරන්න

03 | 3. Questions data සකස් කරන්න

04 | 4. Logic functions පරීක්ෂා කරන්න

05 | 5. DOM buttons සම්බන්ධ කරන්න

06 | 6. Score සහ result පෙන්වන්න

07 | 7. QA කර publish කරන්න

ACTIVITY

මේ පියවර 7 සටහන් පොතේ ලියා ඉදිරියේ checkbox එකක් ඇඳගන්න.

UNIT 05 • LITTLE STAR QUIZ

Folder structure – VS Code Explorer

ZIP එක Extract කර VS Code තුළ Project folder එක විවෘත කරන්න.

```
Little-Star-Quiz/
├── index.html
├── css/
│   └── style.css
├── js/
│   ├── questions.js
│   ├── logic.js
│   └── app.js
├── tests/
└── docs/
```

ACTIVITY

හැම file එකේම භූමිකාව වචන දෙකකින් ලියන්න.

UNIT 05 • LITTLE STAR QUIZ

HTML basic page

මුලින්ම HTML document එකක head සහ body සකස් කරමු.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```html
<!DOCTYPE html>
<html lang="si">
<head>
  <meta charset="UTF-8">
  <title>Little Star Quiz</title>
</head>
<body>
  <h1>JavaScript Quiz</h1>
</body>
</html>
```

EXPECTED RESULT

Chrome tab: Little Star Quiz; heading: JavaScript Quiz

TRY IT YOURSELF

ඉහත code index.html ලෙස save කරන්න.

මතක තබාගන්න: Project ZIP හි full index.html file එකත් බලන්න.

UNIT 05 • LITTLE STAR QUIZ

CSS file එක link කරන්න

පිටුවේ හැඩය HTML වලින් වෙන් කර තබමු.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```html
<head>
  <link rel="stylesheet" href="css/style.css">
</head>
```

EXPECTED RESULT

CSS file එකේ styles browser එකට ලැබේ.

TRY IT YOURSELF

href path එක වෙනස් කළොත් style නැතිවෙනද?

මතක තබාගන්න: Folder path වල spelling ඉතාම වැදගත්.

UNIT 05 • LITTLE STAR QUIZ

JavaScript files link කරන්න

defer භාවිතා කර HTML අවසන් වී පසුව JavaScript ක්‍රියාත්මක වීමට ඉඩ දෙමු.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
<script src="js/questions.js" defer></script>
<script src="js/logic.js" defer></script>
<script src="js/app.js" defer></script>
```

EXPECTED RESULT

questions → logic → app යන අනුපිළිවෙලට scripts run වේ.

TRY IT YOURSELF

app.js file එක නොමැතිනම් Chrome Console එකේ බලන්න.

මතක තබාගන්න: Files එකිනෙක භාවිතා කරන නිසා order එක වෙනස් නොකරන්න.

UNIT 05 • LITTLE STAR QUIZ

Welcome screen එක හදමු

පළමු screen එකේ nickname input එක හා start button එක තබමු.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```html
<section id="welcome">
  <label for="player">Nickname</label>
  <input id="player" placeholder="Little Star">
  <button id="start">Start Quiz</button>
</section>
```

EXPECTED RESULT

Input field + Start Quiz button

TRY IT YOURSELF

Input එක හිස්වම තියලා start කිරීමට ඉඩ දෙන්න.

මතක තබාගන්න: label සහ input id එක එකට ගැළපෙයි.

UNIT 05 • LITTLE STAR QUIZ

Quiz screen එක සකස් කරන්න

Question, answers, score සහ next button එක HTML තුළ තබමු.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```html
<section id="quiz" hidden>
  <p id="progress"></p>
  <p id="score"></p>
  <h2 id="question"></h2>
  <div id="answers"></div>
  <p id="feedback"></p>
  <button id="next" hidden>Next</button>
</section>
```

EXPECTED RESULT

මුලින් hidden; JS මඟින් පෙන්වයි.

TRY IT YOURSELF

next button එක ප්‍රශ්නයට පිළිතුරු දෙන තෙක් hidden තබන්නේ ඇයි?

මතක තබාගන්න: Project ZIP එකේ accessible progress bar එකත් ඇත.

UNIT 05 • LITTLE STAR QUIZ

Result screen එක ලියමු

Quiz අවසන් වූ විට පෙන්වන අවසාන section එක.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```html
<section id="result" hidden>
  <h2>My Result</h2>
  <p id="final-score"></p>
  <p id="message"></p>
  <p id="best"></p>
  <ol id="review"></ol>
  <button id="again">Play Again</button>
</section>
```

EXPECTED RESULT

Final score + Review + Play Again button

TRY IT YOURSELF

score එකට වෙනම id එකක් දීමේ ප්‍රයෝජනය පැහැදිලි කරන්න.

මතක තබාගන්න: ලකුණු හා feedback දරුවන්ට දිරිගන්වන ආකාරයෙන් පෙන්වමු.

UNIT 05 • LITTLE STAR QUIZ

CSS: White design

පිරිසිදු සුදු background එකක්, පැහැදිලි අකුරු සහ focus styles එකතු කරන්න.

VISUAL STUDIO CODE — style.css

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```css
body { background: #fff; color: #173641; }
.panel { border: 2px solid #abc2c5; }
button { background: #087d83; color: white; }
button:focus-visible {
  outline: 3px solid #e0ab2d;
}
```

EXPECTED RESULT

White page + teal buttons

TRY IT YOURSELF

Background එක වෙනස් නොකර border color එක අත්හදා බලන්න.

මතක තබාගන්න: PDF එකේ පිටුවත් සුදු; website එකත් ඉගෙනීමට පහසු සුදු.

UNIT 05 • LITTLE STAR QUIZ

Responsive design

Mobile screen එකක options දෙකක් අසල අසල නොපෙනෙන ලෙස CSS වෙනස් කරමු.

VISUAL STUDIO CODE — style.css

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```css
.answers {
  display: grid;
  grid-template-columns: 1fr 1fr;
}
@media (max-width: 560px) {
  .answers { grid-template-columns: 1fr; }
}
```

EXPECTED RESULT

Desktop: columns 2; mobile: column 1

TRY IT YOURSELF

Chrome DevTools Toggle Device Toolbar භාවිතා කරන්න.

මතක තබාගන්න: Mobile page එකකට horizontal scrolling අවශ්‍ය නොවිය යුතුයි.

UNIT 05 • LITTLE STAR QUIZ

Question data – object එකක්

එක් ප්‍රශ්නයකට question, options සහ correct යන fields තුනක් ඇත.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const question = {
  question: "2 + 2?",
  options: ["2", "3", "4", "5"],
  correct: 2
};
console.log(question.options[2]);
```

EXPECTED RESULT

4

TRY IT YOURSELF

correct = 0 කළොත් කුමක් වෙනස් වේද?

මතක තබාගන්න: Index 0 සිට ගණන් කරන නිසා correct:2 යනු තුන්වන option.

UNIT 05 • LITTLE STAR QUIZ

Questions array එක

ප්‍රශ්න 10ක් වෙන වෙනම files ලෙස නොව එක array එකක තබමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const questions = [
  { question: "2 + 2?", options: ["3","4"], correct: 1 },
  { question: "3 + 3?", options: ["5","6"], correct: 1 }
];
console.log(questions.length);
```

EXPECTED RESULT

2

TRY IT YOURSELF

තවත් question එකක් එකතු කරන්න.

මතක තබාගන්න: Source project එකේ සිංහල ප්‍රශ්න 10ක් සහ options 4 බැගින් ඇත.

UNIT 05 • LITTLE STAR QUIZ

Logic function – හරිද?

Selected answer සහ correct index එක සසඳමු.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function isCorrect(selected, correct) {
  return Number.isInteger(selected) &&
    selected === correct;
}
console.log(isCorrect(2, 2));
console.log(isCorrect(1, 2));
```

EXPECTED RESULT

true සහ false

TRY IT YOURSELF

isCorrect("2", 2) false වන්නේ ඇයි?

මතක තබාගන්න: Project එකේ logic.js මේ function එක ඇතුළත් කරයි.

UNIT 05 • LITTLE STAR QUIZ

Logic function – percentage

ලකුණු ප්‍රතිශතයක් බවට පරිවර්තනය කරමු.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function percentage(score, total) {
  if (!Number.isFinite(score) || total <= 0) return 0;
  return Math.round(score / total * 100);
}
console.log(percentage(8, 10));
```

EXPECTED RESULT

80

TRY IT YOURSELF

percentage(7, 10) සහ percentage(0, 0) පරීක්ෂා කරන්න.

මතක තබාගන්න: සම්පූර්ණ function එක `logic.js` හි තවත් invalid input check එකක් සමඟ ඇත.

UNIT 05 • LITTLE STAR QUIZ

DOM helper function

එකම getElementById නැවත නැවත ලියනවා වෙනුවට helper එකක් හදමු.

VISUAL STUDIO CODE

script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const el = (id) => document.getElementById(id);
const startButton = el("start");
startButton.textContent = "Play Now";
```

EXPECTED RESULT

Start button → Play Now

TRY IT YOURSELF

el("missing") කියන විට ලැබෙන්නේ කුමක්ද?

මතක තබාගන්න: Selector එකෙන් null ලැබුණොත් property access කිරීමට පෙර check කරන්න.

UNIT 05 • LITTLE STAR QUIZ

show() – screen මාරු කරමු

Welcome, Quiz, Result එකවර නොපෙනෙන ලෙස තබමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function show(sectionId) {
  for (const id of ["welcome", "quiz", "result"]) {
    el(id).hidden = id !== sectionId;
  }
}
show("quiz");
```

EXPECTED RESULT

Quiz section පමණක් පෙනේ.

TRY IT YOURSELF

show("result") run කළ විට මොකද වෙන්නේ?

මතක තබාගන්න: Project එකේ show() function එකේ මෙයම භාවිතා කරයි.

UNIT 05 • LITTLE STAR QUIZ

startQuiz() – state reset

Quiz එක පටන්ගන්නා විට score, position, answers හිස් කරමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
let score = 0;
let position = 0;
let selectedAnswers = [];
function startQuiz() {
  score = 0;
  position = 0;
  selectedAnswers = [];
  show("quiz");
  renderQuestion();
}
```

EXPECTED RESULT

Question 1 සහ score 0

TRY IT YOURSELF

Replay කරනවිට score reset නොකළහොත් ඇති ප්‍රශ්නය ලියන්න.

මතක තබාගන්න: State එක app එකේ දැනට තිබෙන තත්ත්වයයි.

UNIT 05 • LITTLE STAR QUIZ

renderQuestion() – ප්‍රශ්නය පෙන්වන්න

Array එකේ දැනට තිබෙන question object එකේ text එක browser එකට පෙන්වමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
function renderQuestion() {
  const item = questions[position];
  el("question").textContent = item.question;
  el("progress").textContent =
    `Question ${position + 1} / ${questions.length}`;
}
```

EXPECTED RESULT

Question 1 / 10 + question text

TRY IT YOURSELF

position 0 වුණත් display එකේ 1 වන්නේ ඇයි?

මතක තබාගන්න: Source app.js එකේ answers buttons ද නිර්මාණය කරයි.

UNIT 05 • LITTLE STAR QUIZ

Options buttons හතර

HTML තුළ buttons 4ක් copy නොකර options array එකෙන් build කරමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const box = el("answers");
box.replaceChildren();
item.options.forEach((option, index) => {
  const button = document.createElement("button");
  button.textContent = `${index + 1}. ${option}`;
  button.addEventListener("click", () => chooseAnswer(index));
  box.appendChild(button);
});
```

EXPECTED RESULT

Quiz question එකට buttons හතරක්

TRY IT YOURSELF

button index එක click handler එකට යන්නේ කොහොමද?

මතක තබාගන්න: මේ block එක renderQuestion() ඇතුළත ඇත.

UNIT 05 • LITTLE STAR QUIZ

chooseAnswer() – ලකුණු දීම

User තෝරාගත් answer එක correct index එක සමඟ සසඳා score එක වැඩි කරමු.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function chooseAnswer(index) {
  const item = questions[position];
  const correct = logic.isCorrect(index, item.correct);
  if (correct) score += 1;
  el("score").textContent = `Score: ${score}`;
}
```

EXPECTED RESULT

Correct → score +1; wrong → +0

TRY IT YOURSELF

එකම answer button දෙවරක් click කළොත් කුමක්ද?

මතක තබාගන්න: මේක teaching extract එකක්; සම්පූර්ණ app එක double scoring වැළැක්වෙයි.

UNIT 05 • LITTLE STAR QUIZ

Double scoring වැළැක්වීම

පිළිතුරක් තෝරා අවසන් වූ පසු එම question එකට යළි score දිය නොයුතුයි.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
if (selectedAnswers.length !== position) return;
selectedAnswers.push(index);
buttons.forEach(button => {
  button.disabled = true;
});
```

EXPECTED RESULT

එක ප්‍රශ්නයට එකම පිළිතුරක්; buttons disable වේ.

TRY IT YOURSELF

QA test TC05 එකෙන් මෙය පරීක්ෂා කරමු.

මතක තබාගන්න: මෙය chooseAnswer() function එක තුළ ඇති guard එකකි.

UNIT 05 • LITTLE STAR QUIZ

හරි පිළිතුර පාටින් පෙන්වමු

Correct answer එකට CSS class එකක් දෙමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
buttons.forEach((button, i) => {
  if (i === item.correct) button.classList.add("correct");
  if (i === index && !correct) button.classList.add("wrong");
});
```

EXPECTED RESULT

Correct = green border; Wrong = orange border

TRY IT YOURSELF

වර්ණ නොපෙනෙන අයටත් feedback text එක පෙන්විය යුත්තේ ඇයි?

මතක තබාගන්න: බෙහෙවින් පාටට පමණක් නොව වචනවලටත් භාර දෙමු.

UNIT 05 • LITTLE STAR QUIZ

Next button – ඉදිරියට යමු

User පිළිතුරක් තෝරාගත් පසු Next button එක පෙන්වමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function nextQuestion() {
  position += 1;
  if (position === questions.length) {
    finishQuiz();
  } else {
    renderQuestion();
  }
}
```

EXPECTED RESULT

Question 1 → 2; Last → Result

TRY IT YOURSELF

condition එක `>` කළොත් ප්‍රශ්නයක් මග හැරෙයිද?

මතක තබාගන්න: Complete project එකේ click guard එකත් ඇත.

UNIT 05 • LITTLE STAR QUIZ

Result screen එකට score

අවසන් score එක 10න් කොපමණදැයි පෙන්වන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function finishQuiz() {
  show("result");
  el("final-score").textContent =
    `${score} / ${questions.length} (` +
    logic.percentage(score, questions.length) + "%)";
}
```

EXPECTED RESULT

8 / 10 (80%) වගේ result එකක්

TRY IT YOURSELF

Result screen එකේ තව පෙන්විය හැකි දත්ත 2ක් ලියන්න.

මතක තබාගන්න: Full app එකේ encouraging message එකක් ද ඇත.

UNIT 05 • LITTLE STAR QUIZ

Best score – පරණ ලකුණු

නව ලකුණු පරණ හොඳම ලකුණු වලට වඩා ඉහළද බලන්න.

VISUAL STUDIO CODE script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
function higherScore(current, oldBest) {
  return Math.max(current, oldBest);
}
console.log(higherScore(6, 9));
console.log(higherScore(10, 9));
```

EXPECTED RESULT

9, 10

TRY IT YOURSELF

4 හා 8 සසඳන result එක ගන්න.

මතක තබාගන්න: Best score එක browser එකටම සීමා වේ; account එකක් හෝ leaderboard එකක් නොවෙයි.

UNIT 05 • LITTLE STAR QUIZ

Review answers – ඉගෙනගමු

Quiz අවසානයේ ප්‍රශ්න, user තෝරාගත් පිළිතුරු සහ නිවැරදි පිළිතුරු පෙන්වන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
questions.forEach((item, index) => {
  const li = document.createElement("li");
  const correct = item.options[item.correct];
  const chosen = item.options[selectedAnswers[index]];
  li.textContent = `You: ${chosen} | Correct: ${correct}`;
  review.appendChild(li);
});
```

EXPECTED RESULT

Result page → question-by-question review

TRY IT YOURSELF

මතක තබාගැනීමට review එක වැදගත් ඇයි?

මතක තබාගන්න: Full project එකේ review variable එක ol element එකක් වේ.

UNIT 05 • LITTLE STAR QUIZ

Replay – නැවත කරන්න

Play Again click කළ විට startQuiz() run කර අලුතින් පටන්ගනිමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
el("again").addEventListener("click", startQuiz);
```

EXPECTED RESULT

Result → Quiz; score නැවත 0

TRY IT YOURSELF

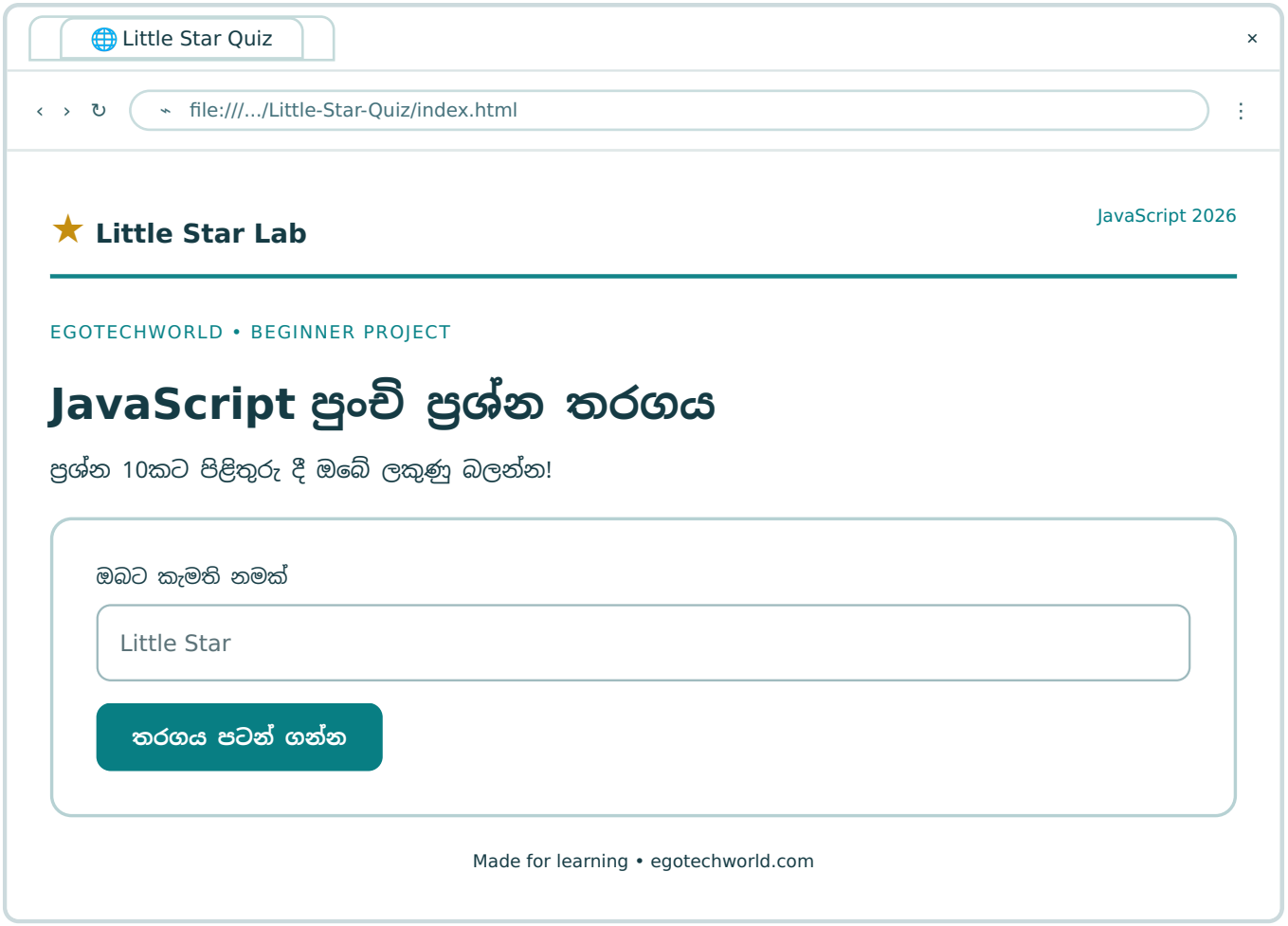
player name එක reset විය යුතුද කියලා user experience එක ගැන සිතන්න.

මතක තබාගන්න: Play again button එකෙන් තවත් ප්‍රශ්න ඇරඹිය හැක.

UNIT 05 • LITTLE STAR QUIZ

Chrome Preview – Welcome

ඉහත code files එකතු වූ පසු welcome screen එක මෙහෙම පෙනෙයි.



පැහැදිලි කිරීම සඳහා නිර්මාණය කළ Chrome-style preview එකක්; browser version අනුව තිරය වෙනස් විය හැක.

- White background සහ top brand.
- Nickname input හිස්ව තිබුණත් start කරන්න පුළුවන්.
- Start button keyboard Tab/Enter මඟින් භාවිතා කළ හැක.

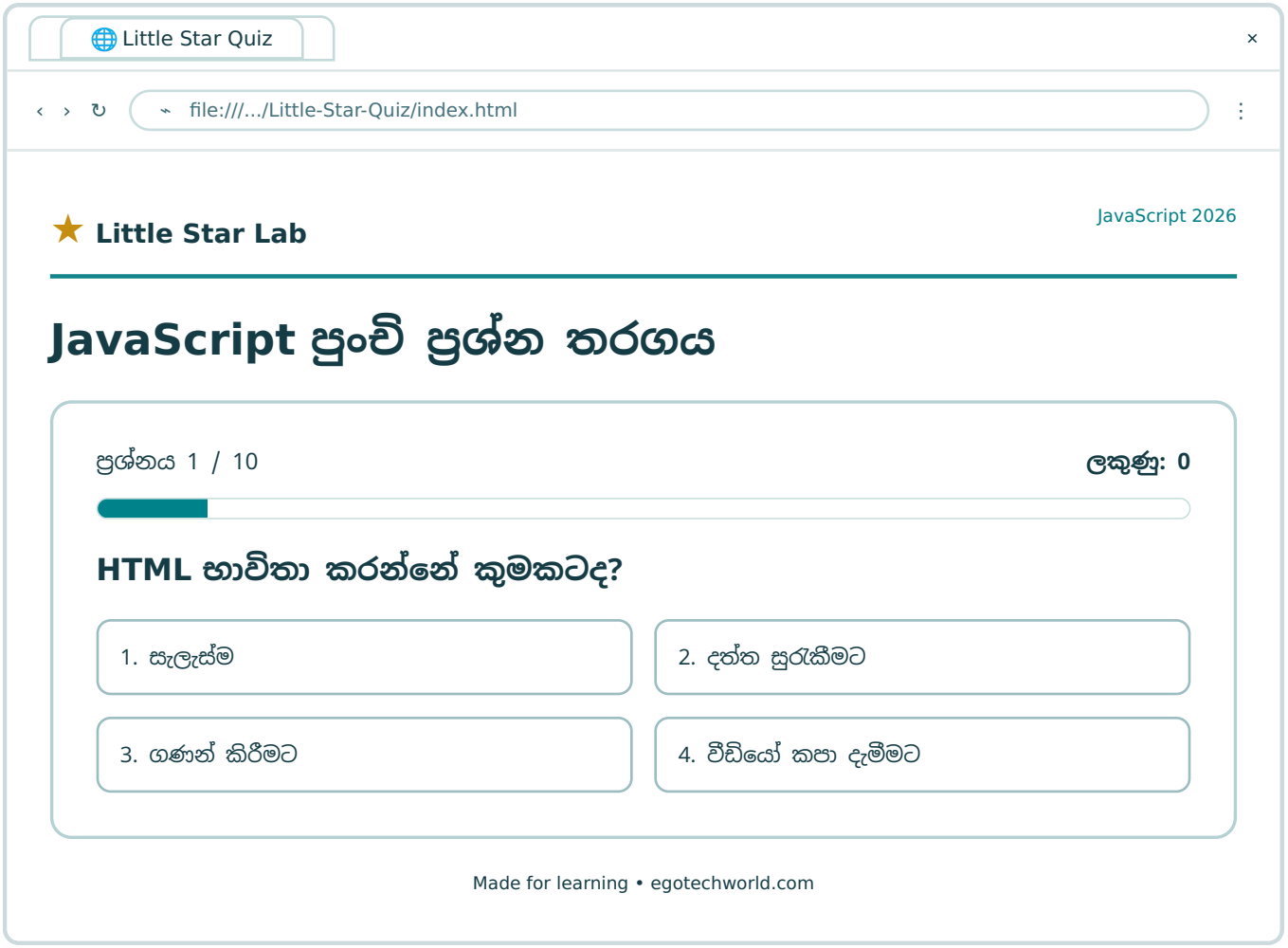
TRY IT

Source ZIP → index.html double-click කර මේ screen එක බලන්න.

UNIT 05 • LITTLE STAR QUIZ

Chrome Preview – Question

වැඩ කරන quiz screen එකේ question number, score සහ answer choices පෙන්වේ.



පැහැදිලි කිරීම සඳහා නිර්මාණය කළ Chrome-style preview එකක්; browser version අනුව තිරය වෙනස් විය හැක.

- හරි පිළිතුරට ලකුණු එකක් එකතු වේ.
- පිළිතුර දුන් පසු buttons disable වේ.
- Next මඟින් ඊළඟ ප්‍රශ්නයට යන්න.

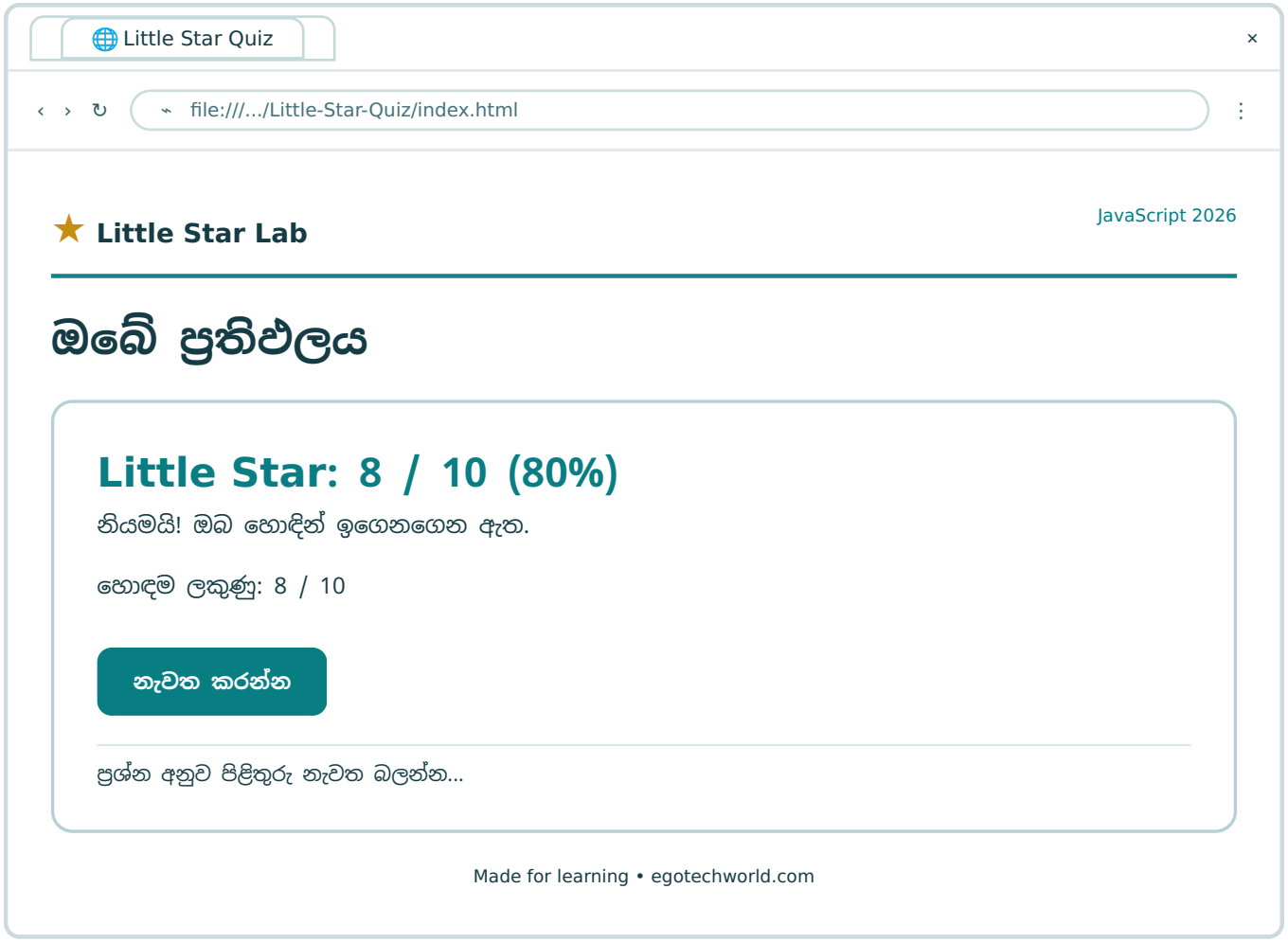
TRY IT

ඔබට කුමන ප්‍රශ්නයේදී score එක 1 වෙයිද කියා හිතන්න.

UNIT 05 • LITTLE STAR QUIZ

Chrome Preview – Result

ප්‍රශ්න 10 අවසන් වූ පසු result එක සහ review ලැයිස්තුව පෙන්වේ.



පැහැදිලි කිරීම සඳහා නිර්මාණය කළ Chrome-style preview එකක්; browser version අනුව තිරය වෙනස් විය හැක.

- Result percentage එක Math.round භාවිතයෙන් ගණනය කරයි.
- Best score මේ browser එක තුළම සුරැකේ.
- Clear best score මඟින් එය ඉවත් කළ හැක.

TRY IT

Play Again කර පහළ score එකක් ලබාගත්තත් best එක වෙනස් නොවන්නේ ඇයි?

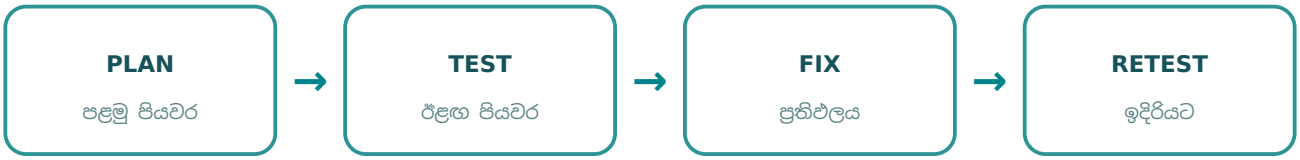
UNIT 06 • QA AND RELEASE

Testing කියන්නේ මොකක්ද?

Code එක වැඩ කරනවා වගේ පෙනුණා කියලා අවසන් නැහැ. අපි ඒක පරීක්ෂා කළ යුතුයි.

- Unit Testing: කුඩා function එකක් පරීක්ෂා කිරීම.
- Integration Testing: files සහ components එකට වැඩ කරනවාද බලන්න.
- UI Testing: User click කළ විට නිවැරදි response එනවාද?
- Regression: අලුත් වෙනසක් පැරණි දේ කඩනවාද?

VISUAL LEARNING MAP



ACTIVITY

ඔබේ quiz project එකට test cases 3ක් ලියන්න.

UNIT 06 • QA AND RELEASE

Manual Test Case 01–05

වැඩ කරන app එක Chrome තුළ විවෘත කර පහත පරීක්ෂණ ක්‍රියාත්මක කරන්න.

01	TC01: index.html විවෘත → Welcome visible
02	TC02: nickname හිස්ව start → Quiz visible
03	TC03: Correct answer click → score +1
04	TC04: Wrong answer click → score වෙනස් නොවේ
05	TC05: Button දෙවරක් click → score එක වරක් පමණයි

ACTIVITY

PASS / FAIL / NOT TESTED ලකුණු කර ලියාගන්න.

UNIT 06 • QA AND RELEASE

Manual Test Case 06–10

අවසාන ප්‍රතිඵලයත් පරීක්ෂා කිරීම අත්‍යවශ්‍යයි.

01	TC06: Question 10 පසු Result screen
02	TC07: Play again → Question 1; score 0
03	TC08: New lower score → best score පරණමයි
04	TC09: Clear best → saved number ඉවත් වේ
05	TC10: 400px width → answer choices එක column එක

ACTIVITY

ඔබේ screen එක 400px කිරීමට Chrome Device Toolbar භාවිතා කරන්න.

UNIT 06 • QA AND RELEASE

Automated test එකක්

Pure function එකක් Node.js මඟින් test කරන්න පුළුවන්.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const assert = require("node:assert/strict");
function add(a, b) { return a + b; }
assert.equal(add(2, 3), 5);
console.log("TEST PASS");
```

EXPECTED RESULT

Terminal: TEST PASS

TRY IT YOURSELF

Expected answer 5 වෙනුවට 6 කර test fail කර බලන්න.

මතක තබාගන්න: Source ZIP තුළ Node.js test suite එකක් ඇත.

UNIT 06 • QA AND RELEASE

Node test command

Source ZIP එකේ test file එක තිබේ. Node.js installed නම් පමණක් ධාවනය කරන්න.

WINDOWS CMD — Terminal

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
node --test tests/logic.test.cjs
```

EXPECTED RESULT

Terminal: 8 tests; 8 pass

TRY IT YOURSELF

Unit tests වලින් browser click behavior සියල්ල පරීක්ෂා වනවාද?

මතක තබාගන්න: Browser E2E QA වෙනම කිරීමට අවශ්‍යයි.

UNIT 06 • QA AND RELEASE

Console debugging

දෝෂයක් ආවොත් console.log තාවකාලිකව එකතු කර values බලන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
console.log("position:", position);
console.log("score:", score);
console.log("answers:", selectedAnswers);
```

EXPECTED RESULT

Current state values console එකේ පෙනේ.

TRY IT YOURSELF

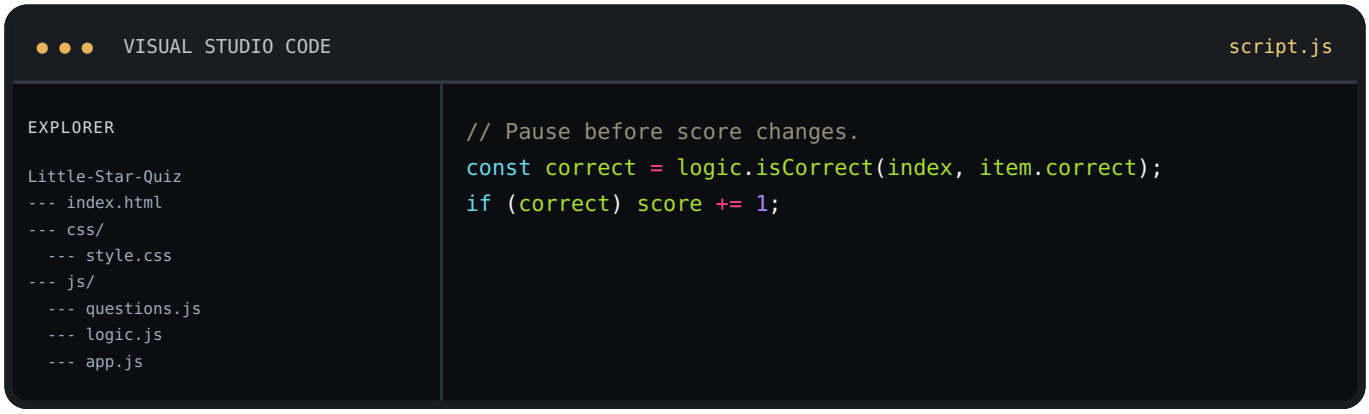
Value එක unexpected වූ තැන සොයන්න.

මතක තබාගන්න: Debug messages පසුව ඉවත් කරන්න; රහස් දත්ත print කරන්න එපා.

UNIT 06 • QA AND RELEASE

Breakpoints

Chrome Sources panel තුළ line එකක breakpoint එකක් දමා code එක නතර කර inspect කරන්න.



EXPECTED RESULT

Pause → inspect index, item.correct, score

TRY IT YOURSELF

index එක සහ correct index එක අන්‍යෝන්‍යව සසඳන්න.

මතක තබාගන්න: Screenshots illustrative; Chrome versions අනුව panels වෙනස් විය හැක.

UNIT 06 • QA AND RELEASE

Bug report එකක් ලියමු

QA engineer කෙනෙකුට තේරෙන ලෙස error එක විස්තර කරන්න.

01 | ID: BUG-001

02 | Title: Score increases twice on double click

03 | Steps: Open → Start → answer double click

04 | Expected: score +1

05 | Actual: score +2

06 | Priority: Medium

07 | Fix: Lock answered state and disable buttons

ACTIVITY

ඔබට හමු වූ තවත් issue එකකට bug report එකක් ලියන්න.

UNIT 06 • QA AND RELEASE

Accessibility – හැමෝටම website

අයෙකුට වර්ණ නොපෙනීම, mouse භාවිතා කළ නොහැකි වීම හෝ කුඩා තිරයක් තිබීම සාමාන්‍යයි.

- Buttons සැබෑ HTML button elements ලෙස සාදන්න.
- Input field එකට label භාවිතා කරන්න.
- Success/error සඳහා පාටට අමතරව වචන දෙන්න.
- Keyboard focus පෙනෙන outline එකක් CSS තුළ තබන්න.
- Screen readers සඳහා status messages aria-live සමඟ තබාගන්න.

VISUAL LEARNING MAP



ACTIVITY

Tab key එකෙන් පමණක් Welcome → Quiz කර බලන්න.

UNIT 06 • QA AND RELEASE

Security සහ Privacy – දරුවන්ට ඉතා වැදගත්

JavaScript browser එකට යවන නිසා source code එක user ට දැකගන්න පුළුවන්.

- Passwords, API secrets, private IDs JavaScript files තුළ තබන්න එපා.
- පිටතින් ආ text එක HTML ලෙස execute නොකරන්න.
- `textContent` සහ `createElement` භාවිතා කරන්න.
- localStorage යනු secret vault එකක් නොවෙයි.
- සැබෑ login/database app එකකට server-side security අවශ්‍යයි.

VISUAL LEARNING MAP



ACTIVITY

Project එක තුළ textContent භාවිතා කරන line දෙකක් සොයන්න.

UNIT 06 • QA AND RELEASE

GitHub Pages වෙත publish කරන්න

මෙම quiz එක static website එකක් නිසා GitHub Pages තුළ publish කළ හැක. වැඩිහිටි/ගිණුම් වයස් නීති පිළිපදින්න.

01 | GitHub repository එකක් සාදන්න

02 | `index.html`, css/, js/ files upload කරන්න

03 | Repository Settings → Pages වෙත යන්න

04 | Source: Deploy from a branch තෝරන්න

05 | Branch: main, folder: / (root)

06 | Save → Pages URL එක ලබාගන්න

ACTIVITY

පෞද්ගලික දත්ත ZIP එකේ නැති බව publish කිරීමට පෙර check කරන්න.

UNIT 06 • QA AND RELEASE

Git commands (optional)

Git installed නම් local folder එක GitHub repository එකට යැවිය හැක.

WINDOWS CMD — Terminal

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
git init
git add .
git commit -m "Build my first JavaScript quiz"
git branch -M main
git remote add origin YOUR_REPO_URL
git push -u origin main
```

EXPECTED RESULT

GitHub repository එකේ project files දිස්වේ.

TRY IT YOURSELF

YOUR_REPO_URL වෙනුවට ඔබේ සැබෑ repository URL එක යොදන්න.

මතක තබාගන්න: GitHub authentication අවශ්‍ය වනු ඇත; passwords code හෝ screenshot වල දමන්න එපා.

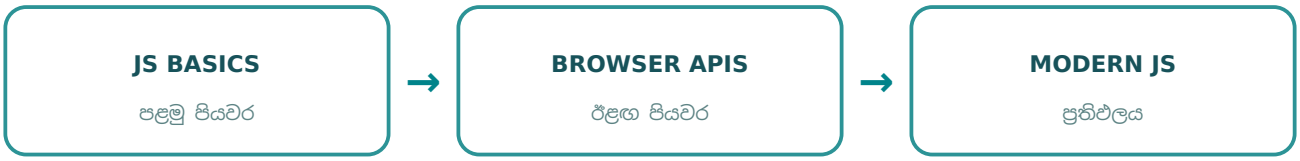
UNIT 07 • 2026 AND BEYOND

JavaScript 2026 කියන්නේ?

JavaScript පාලනය කරන භාෂා සම්මතය ECMAScript ලෙස හඳුන්වයි. 2026 වර්ෂයේ නිල ECMA-262 17th edition එක නිකුත් කර ඇත.

- Beginner කෙනෙකුට අදත් let, const, functions, arrays ඉතා වැදගත්.
- Modern browser තුළ DOM, events, fetch සහ modules ඉගෙනගන්න.
- 2026 features නොමැති browser වලට ගැළපීම ගැන සැලකිලිමත් වෙන්න.
- Source: Ecma International ECMA-262 (June 2026) සහ MDN JavaScript Guide.

VISUAL LEARNING MAP



ACTIVITY

JavaScript හා ECMAScript වචන දෙක අතර වෙනස සටහන් කරන්න.

UNIT 07 • 2026 AND BEYOND

Promise සරල අදහස

කාලයක් ගන්නා කාර්යයක් අවසන් වූ පසු result එකක් ලබාදීමට Promise භාවිතා කළ හැක.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
const promise = Promise.resolve("Ready!");
promise.then((value) => {
  console.log(value);
});
```

EXPECTED RESULT

Ready!

TRY IT YOURSELF

Promise එකට "Finished!" message එකක් යොදන්න.

මතක තබාගන්න: Asynchronous code භාවිතයෙන් browser එක දිගටම ප්‍රතිචාර දිය හැක.

UNIT 07 • 2026 AND BEYOND

async / await

Promise එකක් සම්බන්ධ code එක කියවීමට පහසු ලෙස async/await භාවිතා කරමු.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
async function sayReady() {
  const result = await Promise.resolve("Ready!");
  console.log(result);
}
sayReady();
```

EXPECTED RESULT

Ready!

TRY IT YOURSELF

result එකට number එකක් return කරන්න.

මතක තබාගන්න: await භාවිතය සාමාන්‍යයෙන් async function එකක් තුළ වේ.

UNIT 07 • 2026 AND BEYOND

fetch – API දත්ත ඉල්ලමු

Server එකෙන් JSON ගන්න fetch භාවිතා කළ හැකි නමුත් offline website එකට network අවශ්‍ය නැහැ.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```js
async function getExample() {
  const response = await fetch("data.json");
  if (!response.ok) throw new Error("HTTP error");
  const data = await response.json();
  console.log(data);
}
getExample().catch(console.error);
```

EXPECTED RESULT

Server එකක /data.json තිබුණොත් JSON ලැබේ.

TRY IT YOURSELF

මෙය run කරන්න local web server එකක් හෝ hosted site එකක් අවශ්‍යයි.

මතක තබාගන්න: file:// ලෙස direct open කරනවිට fetch ප්‍රතික්ෂේප විය හැක.

UNIT 07 • 2026 AND BEYOND

ES Modules import/export

Large projects වල code එක වෙන් කර තබාගැනීමට modules උපකාරීයි.



EXPECTED RESULT

5 (module-enabled environment එකක)

TRY IT YOURSELF

HTML එකේ <script type="module"> භාවිතා කරන්න.

මතක තබාගන්න: Browser modules local `file://` වලින් නොව local server එකකින් test කරන්න.

UNIT 07 • 2026 AND BEYOND

Frameworks – React, Vue, Angular

JavaScript basics ඉගෙනගත් පසු විශාල websites සඳහා frameworks/library භාවිතා කළ හැක.

- React = reusable UI components සඳහා ජනප්‍රිය library එකක්.
- Vue සහ Angular = UI development සඳහා වෙනත් මෙවලම්.
- Node.js = browser එකෙන් පිටත JavaScript ධාවනය කරයි.
- Backend + database = සැබෑ users දත්ත සුරකින applications සඳහා.
- පළමුව plain JavaScript හොඳින් ඉගෙනගත් විට frameworks ඉගෙනීම පහසු වේ.

VISUAL LEARNING MAP



ACTIVITY

Quiz app එකේ reusable component එකක් විය හැකි දෙයක් ලියන්න.

UNIT 07 • 2026 AND BEYOND

AI tools භාවිතා කරන ආකාරය

2026 දී AI coding assistants තිබුණත් code එක තේරුම්ගැනීම සහ පරීක්ෂා කිරීම ඔබගේ වගකීමයි.

- AI එකෙන් "මේ line එක තේරුම් කර දෙන්න" ලෙස අහන්න.
- AI code එක copy කළත් test cases run කරන්න.
- Password, සැබෑ customer data, private project files නොබෙදාගන්න.
- AI නිවැරදි නැති code හෝ හෙළි නොකළ අවදානම් යෝජනා කළ හැක.
- ඔබම build → test → explain කළ විට සැබෑ දැනුම ලැබේ.

VISUAL LEARNING MAP



ACTIVITY

AI එකට දිය හැකි ආරක්ෂිත ඉගෙනීමේ ප්‍රශ්න 2ක් ලියන්න.

අවසන් අභ්‍යාස

දින 7ක ඉගෙනීමේ සැලැස්ම

අභ්‍යාස කුඩා කොටස්වලට බෙදුවොත් හැමදාම ඉගෙනගැනීම පහසුයි.

01	Day 1 – HTML, CSS, console.log සහ variables
02	Day 2 – Numbers, string, if/else, loops
03	Day 3 – Functions, arrays, objects
04	Day 4 – DOM, events, inputs
05	Day 5 – Quiz HTML/CSS හා questions data
06	Day 6 – Score, review, localStorage
07	Day 7 – Testing, bug fixes, GitHub Pages

ACTIVITY

සෑම දවසක් අවසානයේ ඔබ ඉගෙනගත් කරුණු 3ක් ලියන්න.

7-DAY PRACTICE

දවස 1 – ඔබේ පළමු message එක

මෙම දවසේ අරමුණ තේරුම්ගෙන කුඩා කාර්ය තුනම සම්පූර්ණ කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
console.log("I am learning JS!");
```

- Chrome Console විවෘත කරන්න
- Name variable එකක් සාදන්න
- Hello message එක වෙනස් කරන්න

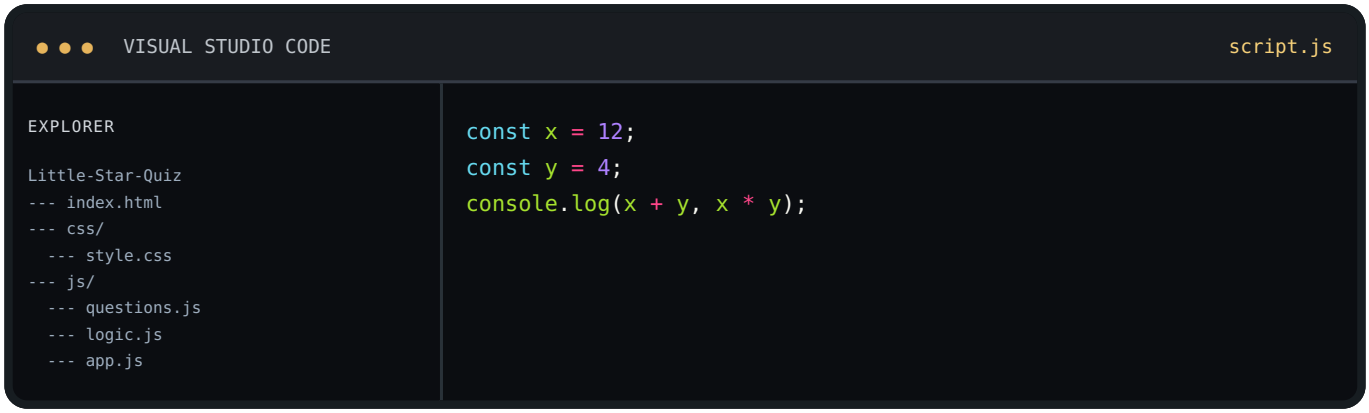
JOURNAL

අවසන් වූ පසු “මට අද හැකි දේ” ලෙස එක් වාක්‍යයක් ලියන්න.

7-DAY PRACTICE

දවස 2 – කුඩා calculator එක

මෙම දවසේ අරමුණ තේරුම්ගෙන කුඩා කාර්ය තුනම සම්පූර්ණ කරන්න.



- Numbers 2ක් යොදන්න
- if score >= 5 කොන්දේසිය ලියන්න
- for loop එකකින් 1 සිට 5 දක්වා print කරන්න

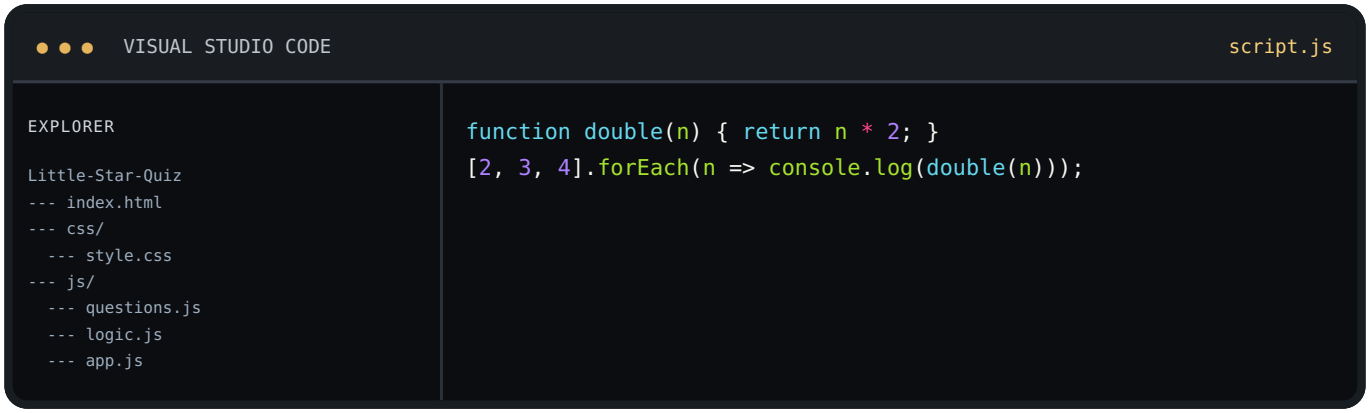
JOURNAL

අවසන් වූ පසු “මට අද හැකි දේ” ලෙස එක් වාක්‍යයක් ලියන්න.

7-DAY PRACTICE

දවස 3 – list සහ functions

මෙම දවසේ අරමුණ තේරුම්ගෙන කුඩා කාර්ය තුනම සම්පූර්ණ කරන්න.



- Function parameters කියන්න
- Array index 0 ගැන කියන්න
- Object එකක name හා score සුරකින්න

JOURNAL

අවසන් වූ පසු “මට අද හැකි දේ” ලෙස එක් වාක්‍යයක් ලියන්න.

7-DAY PRACTICE

දවස 4 – DOM සහ button

මෙම දවසේ අරමුණ තේරුම්ගෙන කුඩා කාර්ය තුනම සම්පූර්ණ කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
document.getElementById("start")
  .addEventListener("click", startQuiz);
```

- id එකෙන් element සොයන්න
- Button click event එකක් හදන්න
- textContent යොදා output වෙනස් කරන්න

JOURNAL

අවසන් වූ පසු “මට අද හැකි දේ” ලෙස එක් වාක්‍යයක් ලියන්න.

7-DAY PRACTICE

දවස 5 – Quiz screen

මෙම දවසේ අරමුණ තේරුම්ගෙන කුඩා කාර්ය තුනම සම්පූර්ණ කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
const first = window.QuizData[0];
console.log(first.question);
```

- index.html විවෘත කරන්න
- css/style.css change කරන්න
- questions.js තුළ නව ප්‍රශ්නයක් ලියන්න

JOURNAL

අවසන් වූ පසු “මට අද හැකි දේ” ලෙස එක් වාක්‍යයක් ලියන්න.

7-DAY PRACTICE

දවස 6 – Score සහ result

මෙම දවසේ අරමුණ තේරුම්ගෙන කුඩා කාර්ය තුනම සම්පූර්ණ කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
console.log(window.QuizLogic.percentage(8, 10));
```

- Correct answer එක test කරන්න
- Play Again test කරන්න
- Best score reset test කරන්න

JOURNAL

අවසන් වූ පසු “මට අද හැකි දේ” ලෙස එක් වාක්‍යයක් ලියන්න.

7-DAY PRACTICE

දවස 7 – QA සහ Publish

මෙම දවසේ අරමුණ තේරුම්ගෙන කුඩා කාර්ය තුනම සම්පූර්ණ කරන්න.



- Manual test cases 10ක් කර බලන්න
- Unknown bug එකක් report කරන්න
- Website publish කිරීමට පෙර private data check කරන්න

JOURNAL

අවසන් වූ පසු “මට අද හැකි දේ” ලෙස එක් වාක්‍යයක් ලියන්න.

අවසන් අභ්‍යාස

Final Quiz – කොටස 1

පැන්සලක් හෝ පිටුපස සටහන් පොතක් භාවිතා කර පිළිතුරු ලියන්න.

1. HTML වල කාර්යය?

A. structure • B. database • C. network

2. CSS වල කාර්යය?

A. events • B. design • C. math

3. JavaScript වල කාර්යය?

A. interactivity • B. printing only

4. F12 Chrome තුළ බොහෝවිට විවෘත කරන්නේ?

A. DevTools • B. Calculator

CHALLENGE

එක් ප්‍රශ්නයකට ඔබේ වචනවලින් උදාහරණයක් ද එක් කරන්න.

අවසන් අභ්‍යාස

Final Quiz – කොටස 2

පැන්සලක් හෝ පිටුපස සටහන් පොතක් භාවිතා කර පිළිතුරු ලියන්න.

1. let variable අගය වෙනස් කළ හැකිද?

A. ඔව් • B. නැහැ

2. Strict equality?

A. = • B. === • C. =>

3. Array index ආරම්භ වන්නේ?

A. 1 • B. 0

4. Function result ආපසු දෙන්නේ?

A. return • B. margin

CHALLENGE

Code එක අමතක නම් අදාළ lesson පිටුව බලන්න.

අවසන් අභ්‍යාස

Final Quiz – කොටස 3

පැන්සලක් හෝ පිටුපස සටහන් පොතක් භාවිතා කර පිළිතුරු ලියන්න.

1. Click event එකට?

A. addEventListener • B. parseInt

2. Safe text output?

A. textContent • B. eval

3. Storage එකේ passwords තැබීම?

A. සුදුසුයි • B. සුදුසු නැහැ

4. QA කරන එක වැදගත් ඇයි?

A. bugs හඳුනාගැනීමට • B. files සැඟවීමට

CHALLENGE

ඔබගේ quiz project එකේ feature එකක් වෙනස් කර testing නැවත කරන්න.

අවසන් අභ්‍යාස

Answer Key – MCQ 01–03

පිළිතුරු කියවා ඉගෙනගත් හේතුවත් තේරුම්ගන්න. අංකය පමණක් මතක නොතබන්න.

- MCQ 01: let; නැහැ; 52; සසඳන්න.
- MCQ 02: index 0; return; push; key:value.
- MCQ 03: getElementById; textContent; addEventListener; නැහැ.
- අවසාන Quiz 1: structure, design, interactivity, DevTools.
- අවසාන Quiz 2: ඔව්, ===, 0, return.
- අවසාන Quiz 3: addEventListener, textContent, නුසුදුසු, bugs සොයාගැනීමට.

VISUAL LEARNING MAP



ACTIVITY

වැරදි වූ ප්‍රශ්නයක් තෝරා පිළිතුර හරි වන්නේ ඇයිද කියා පැහැදිලි කරන්න.

අවසන් අභ්‍යාස

Final Project Checklist

ඔබගේ website එක release කිරීමට පෙර මේ ලැයිස්තුවෙන් පරීක්ෂා කරන්න.

- ☐ index.html Chrome එකේ විවෘත වේ.
- ☐ Start button ක්‍රියා කරයි; nickname optional.
- ☐ Questions 10ම දිස්වේ; button 4 බැගින් ඇත.
- ☐ Double click කළත් score දෙවරක් නොවැඩේ.
- ☐ Result page සහ correct answers review පෙන්වේ.
- ☐ Play Again සහ Clear Best ක්‍රියා කරයි.
- ☐ Mobile viewport හොඳින් පෙනේ.
- ☐ Privacy note එක කියවා data සුරක්ෂිතද බලන්න.

VISUAL LEARNING MAP



ACTIVITY

Fail වූ item එකක් තිබුණොත් fix කර නැවත පරීක්ෂා කරන්න.

COURSE COMPLETION

පුංචි JavaScript Developer

මෙම පාඩම් කියවා, Little Star Quiz project එක සම්පූර්ණ කර, ඔබම පරීක්ෂා කළ බව සටහන් කරන්න.

- Student nickname: _____
- Completed date: _____
- Project tested: _____
- Next goal: _____

VISUAL LEARNING MAP



NOTE

මෙය නිල වෘත්තීය සහතිකයක් නොව, ඔබේ ඉගෙනීමේ මතක සටහනකි.

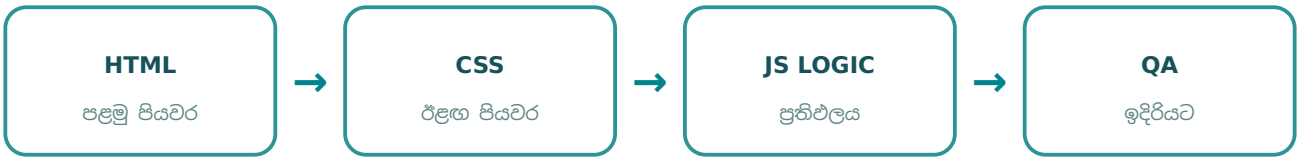
SOURCE CODE APPENDIX

Source files කියවීමේ මාර්ගෝපදේශය

PDF එකේ code කොටස් ඉගෙනීමට හිතාමතා කෙටි කර ඇත. ZIP තුළ සම්පූර්ණ ක්‍රියාකරන files ඇත.

- index.html – website structure සහ link කර ඇති scripts.
- css/style.css – white design, mobile styles සහ focus outlines.
- js/questions.js – 10 data objects සහ options.
- js/logic.js – pure, unit-test කළ හැකි functions.
- js/app.js – DOM/events, score, reviews, localStorage.
- tests/logic.test.cjs – tests 8ක්.
- docs/ – SRS-style plan සහ QA test cases.

VISUAL LEARNING MAP



NOTE

Project එකක source සහ documentation වෙන වෙනම තිබීමේ වාසි කියන්න.

SOURCE CODE APPENDIX

index.html · lines 1–20

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — index.html

```
EXPLORER

Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
1  <!DOCTYPE html>
2  <html lang="si">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1">
6    <title>Little Star JavaScript Quiz</title>
7    <link rel="stylesheet" href="css/style.css">
8    <script src="js/questions.js" defer></script>
9    <script src="js/logic.js" defer></script>
10   <script src="js/app.js" defer></script>
11 </head>
12 <body>
13   <main class="shell">
14     <header class="brand"><span aria-hidden="true">★</span> Little Star Lab
15       <span class="tag">JavaScript 2026</span>
16     </header>
17     <section class="hero" aria-labelledby="title">
18       <p class="eyebrow">EGOTECHWORLD • BEGINNER PROJECT</p>
19       <h1 id="title">JavaScript පුංචි ප්‍රශ්න තරඟය</h1>
20       <p>ප්‍රශ්න 10කට පිළිතුරු දී ඔබේ ලකුණු බලන්න!</p>
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

index.html · lines 21–48

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — index.html

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
21      </section>
22
23      <section id="welcome" class="panel" aria-label="Start quiz">
24        <label for="player">ඔබට කැමති නමක් (අත්‍යවශ්‍ය නැහැ)</label>
25        <input id="player" maxlength="24" placeholder="Little Star" autocomplete="off">
26        <button type="button" id="start">තරගය පටන් ගන්න</button>
27        <p class="hint">සැබෑ නමක් අවශ්‍ය නැහැ. ලකුණු පමණක් මේ browser එකේ මතක
තබාගන්නවා.</p>
28      </section>
29
30      <section id="quiz" class="panel" hidden aria-labelledby="question">
31        <div class="split"><span id="progress">ප්‍රශ්නය 1 / 10</span><span
id="score">ලකුණු: 0</span></div>
32        <progress id="bar" max="10" value="0" aria-label="Quiz progress"></progress>
33        <h2 id="question"></h2>
34        <div id="answers" class="answers" role="group" aria-label="උත්තර තෝරන්න"></
div>
35        <p id="feedback" aria-live="polite"></p>
36        <button type="button" id="next" hidden>ඊළඟ ප්‍රශ්නය</button>
37      </section>
38
39      <section id="result" class="panel" hidden aria-labelledby="result-title">
40        <h2 id="result-title">ඔබේ ප්‍රතිඵලය</h2>
41        <p id="final-score" class="big"></p>
42        <p id="message"></p>
43        <p id="best">මෙතෙක් හොඳම ලකුණු:</p>
44        <h3>පිළිතුරු නැවත බලමු</h3>
45        <ol id="review"></ol>
46        <div class="actions"><button type="button" id="again">නැවත කරන්න</button>
47          <button type="button" class="secondary" id="forget">හොඳම ලකුණු ඉවත් කරන්න</
button></div>
48      </section>
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

css/style.css · lines 1–19

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — style.css

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
 1  /* Beginner-friendly colors and spacing. */
 2  :root { color-scheme: light; font-family: Arial, "Noto Sans Sinhala", sans-serif; }
 3  * { box-sizing: border-box; }
 4  body { background: #fff; color: #15313d; margin: 0; }
 5  .shell { max-width: 820px; margin: 28px auto; padding: 0 20px; }
 6  .brand { display: flex; justify-content: space-between; align-items: center;
 7    font-size: 18px; font-weight: 800; border-bottom: 3px solid #107c80; padding: 13px
0; }
 8  .brand span:first-child { color: #d69a09; }
 9  .tag { color: #126d74; font-size: 12px; }
10  .hero { padding: 32px 0 18px; }
11  .eyebrow { font-size: 11px; font-weight: bold; letter-spacing: 1px; color: #0d797e; }
12  h1 { font-size: clamp(27px, 4vw, 40px); margin: 7px 0 8px; }
13  h2 { font-size: 23px; margin-top: 22px; }
14  p { line-height: 1.8; }
15  .panel { border: 2px solid #b4ccd1; border-radius: 14px; padding: 26px;
16    background: #fff; margin: 14px 0; }
17  label { display: block; font-weight: 700; margin-bottom: 10px; }
18  input { display: block; width: 100%; padding: 13px; border-radius: 9px;
19    border: 2px solid #9fb2b8; font: inherit; margin-bottom: 14px; }
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

css/style.css · lines 20–41

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — style.css

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
20  button { cursor: pointer; border: 0; border-radius: 10px; background: #087d83;
21    color: #fff; font: 700 15px Arial, sans-serif; padding: 14px 18px; margin: 6px 6px
6px 0; }
22  button:hover { background: #075b61; }
23  button:focus-visible, input:focus-visible { outline: 3px solid #e0ab2d; outline-
offset: 3px; }
24  button:disabled { cursor: not-allowed; opacity: .7; }
25  button.secondary { color: #145360; background: white; border: 2px solid #9dbac0; }
26  .hint, footer { color: #52626b; font-size: 13px; }
27  .split { display: flex; justify-content: space-between; gap: 18px; font-weight:
bold; }
28  progress { width: 100%; height: 16px; accent-color: #087d83; margin: 14px 0; }
29  .answers { display: grid; grid-template-columns: 1fr 1fr; gap: 10px; }
30  .answers button { text-align: left; background: #fff; color: #163a40;
31    border: 2px solid #91b4ba; min-height: 66px; margin: 0; }
32  .answers button:hover { border-color: #087d83; background: #e8f7f6; }
33  .answers button.correct { background: #ddf5e9; border-color: #187b52; }
34  .answers button.wrong { background: #fff0ea; border-color: #b75535; }
35  #feedback { min-height: 30px; font-weight: 700; }
36  .big { font-size: 36px; font-weight: 900; color: #087d83; }
37  #review { padding-left: 28px; line-height: 2; }
38  #review li { margin-bottom: 14px; }
39  .actions { margin-top: 18px; }
40  footer { text-align: center; padding: 28px 0; }
41  [hidden] { display: none !important; }
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

js/questions.js · lines 1–11

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
1  // Add questions here. Each correct value is a zero-based option index.
2  window.QuizData = [
3    { question: "HTML භාවිතා කරන්නේ කුමකටද?", options: ["සැලැස්ම", "දත්ත සුරැකීමට",
"ගණන් කිරීමට", "වීඩියෝ කපා දැමීමට"], correct: 0 },
4    { question: "CSS වල ප්‍රධාන කාර්යය කුමක්ද?", options: ["Database", "Page design",
"Email", "Server"], correct: 1 },
5    { question: "JavaScript මඟින් වෙබ් පිටුවකට එකතු කරන්නේ?", options: ["කඩදාසි",
"විදුලිය", "Interactivity", "පින්තූර පමණයි"], correct: 2 },
6    { question: "වෙනස් නොවන අගයක් සඳහා සුදුසු keyword එක?", options: ["change",
"const", "print", "value"], correct: 1 },
7    { question: "5 + 3 හි පිළිතුර කුමක්ද?", options: ["2", "53", "15", "8"], correct: 3 },
8    { question: "if භාවිතා කරන්නේ?", options: ["තීරණ ගැනීමට", "පිටු print කිරීමට",
"Files delete කිරීමට", "Music play කිරීමට"], correct: 0 },
9    { question: "Array එකක දත්ත තබන්නේ?", options: ["එක වචනයක්", "ලැයිස්තුවක්",
"HTML tag එකක්", "Password එකක්"], correct: 1 },
10   { question: "function එකක ප්‍රයෝජනය?", options: ["Code නැවත භාවිතයට", "Monitor
එකට", "Keyboard එකට", "Internet speed"], correct: 0 },
11   { question: "Button එක click කළ විට ක්‍රියාවක් කිරීමට?", options: ["margin",
"<title>", "addEventListener", "border"], correct: 2 },
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

js/questions.js · lines 12–13

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
12    { question: "Console එකට message එකක් පෙන්වන command එක?", options:
["console.log()", "button.style", "HTML()", "math.sum()"], correct: 0 }
13  ];
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

js/logic.js · lines 1–24

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
1  // Pure functions: easy to test without opening the browser.
2  (function (root) {
3    "use strict";
4
5    function percentage(score, total) {
6      if (!Number.isFinite(score) || !Number.isFinite(total) || total <= 0) {
7        return 0;
8      }
9      return Math.round((score / total) * 100);
10   }
11
12   function isCorrect(selected, correct) {
13     return Number.isInteger(selected) && selected === correct;
14   }
15
16   function message(score, total) {
17     const percent = percentage(score, total);
18     if (percent >= 80) return "නියමයි! ඔබ හොඳින් ඉගෙනගෙන ඇත.";
19     if (percent >= 50) return "හොඳ උත්සාහයක්! තව පුහුණු වෙමු.";
20     return "කමක් නැහැ! පාඩම් නැවත බලලා උත්සාහ කරන්න.";
21   }
22
23   function higherScore(current, oldBest) {
24     return Math.max(current, oldBest);
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

js/logic.js · lines 25–28

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
25    }
26
27    root.QuizLogic = { percentage, isCorrect, message, higherScore };
28  })(globalThis);
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

js/app.js · lines 1–30

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
1  // This file connects the HTML page to our quiz logic.
2  "use strict";
3  const questions = window.QuizData;
4  const logic = window.QuizLogic;
5  const el = (id) => document.getElementById(id);
6  const storageKey = "little-star-best-score-v1";
7
8  let position = 0;
9  let score = 0;
10 let selectedAnswers = [];
11 let player = "Little Star";
12
13 function show(sectionId) {
14   for (const id of ["welcome", "quiz", "result"]) {
15     el(id).hidden = id !== sectionId;
16   }
17 }
18
19 function startQuiz() {
20   // Never require a real name from a child.
21   player = el("player").value.trim() || "Little Star";
22   position = 0;
23   score = 0;
24   selectedAnswers = [];
25   show("quiz");
26   renderQuestion();
27 }
28
29 function renderQuestion() {
30   const item = questions[position];
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

js/app.js · lines 31–60

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
31    el("progress").textContent = `ප්‍රශ්නය ${position + 1} / ${questions.length}`;
32    el("score").textContent = `ලකුණු: ${score}`;
33    el("bar").max = questions.length;
34    el("bar").value = position;
35    el("question").textContent = item.question;
36    el("feedback").textContent = "";
37    el("next").hidden = true;
38    const box = el("answers");
39    box.replaceChildren();
40
41    item.options.forEach((option, index) => {
42      const button = document.createElement("button");
43      button.type = "button";
44      button.textContent = `${index + 1}. ${option}`;
45      button.addEventListener("click", () => chooseAnswer(index));
46      box.appendChild(button);
47    });
48  }
49
50  function chooseAnswer(index) {
51    // One answer per question: stop double scoring.
52    if (selectedAnswers.length !== position) return;
53    const item = questions[position];
54    const correct = logic.isCorrect(index, item.correct);
55    selectedAnswers.push(index);
56    if (correct) score += 1;
57    el("score").textContent = `ලකුණු: ${score}`;
58    const buttons = el("answers").querySelectorAll("button");
59    buttons.forEach((button, i) => {
60      button.disabled = true;
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

js/app.js · lines 61–90

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
61      if (i === item.correct) button.classList.add("correct");
62      if (i === index && !correct) button.classList.add("wrong");
63    });
64    el("feedback").textContent = correct
65      ? "✓ නිවැරදියි!" : "උත්සාහයට ස්තූතියි. හරි පිළිතුර හරිත පාටින් පෙන්වලා ඇත.";
66    el("next").textContent = position === questions.length - 1
67      ? "ප්‍රතිඵල බලන්න" : "ඊළඟ ප්‍රශ්නය";
68    el("next").hidden = false;
69    el("next").focus();
70  }
71
72  function nextQuestion() {
73    if (selectedAnswers.length !== position + 1) return;
74    position += 1;
75    if (position === questions.length) finishQuiz();
76    else renderQuestion();
77  }
78
79  function getBestScore() {
80    try {
81      const saved = Number(localStorage.getItem(storageKey));
82      return Number.isFinite(saved) && saved >= 0 ? saved : 0;
83    } catch (_error) {
84      return 0; // Storage can be blocked in private browser settings.
85    }
86  }
87
88  function saveBestScore(value) {
89    try {
90      localStorage.setItem(storageKey, String(value));
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

js/app.js · lines 91–120

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
91    } catch (_error) {
92      // App still works without storage.
93    }
94  }
95
96  function finishQuiz() {
97    show("result");
98    const best = logic.higherScore(score, getBestScore());
99    saveBestScore(best);
100   el("final-score").textContent = `${player}: ${score} / ${questions.length} ($
{logic.percentage(score, questions.length)}%)`;
101   el("message").textContent = logic.message(score, questions.length);
102   el("best").textContent = `මේ browser එකේ හොඳම ලකුණු: ${best} / $
{questions.length}`;
103   const list = el("review");
104   list.replaceChildren();
105   questions.forEach((item, index) => {
106     const li = document.createElement("li");
107     const right = item.options[item.correct];
108     const selected = item.options[selectedAnswers[index]];
109     li.textContent = `${item.question} | ඔබ: ${selected} | හරි: ${right}`;
110     list.appendChild(li);
111   });
112 }
113
114 el("start").addEventListener("click", startQuiz);
115 el("next").addEventListener("click", nextQuestion);
116 el("again").addEventListener("click", startQuiz);
117 el("forget").addEventListener("click", () => {
118   try { localStorage.removeItem(storageKey); } catch (_error) {}
119   el("best").textContent = "හොඳම ලකුණු ඉවත් කළා.";
120 });
```

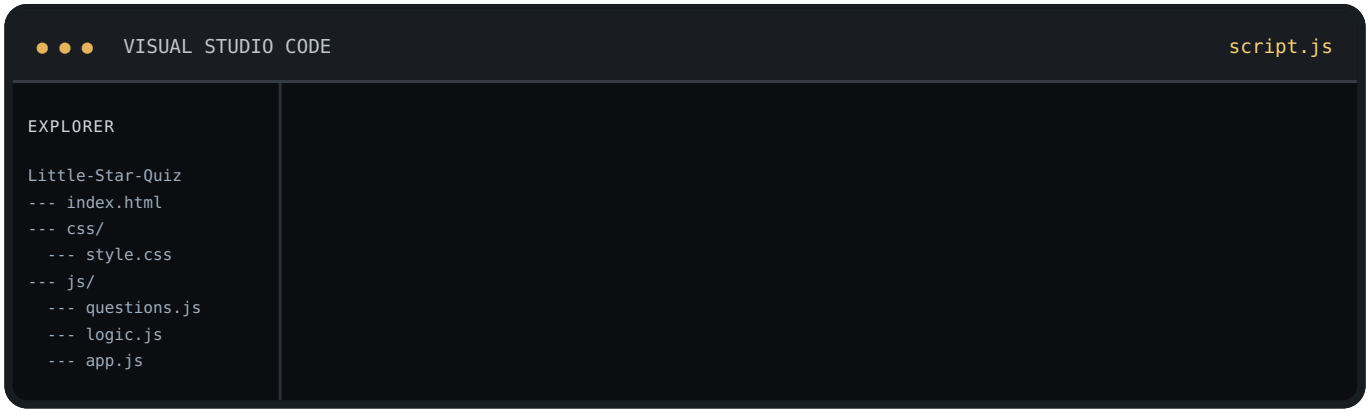
CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

js/app.js · lines 121–120

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.



CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

SOURCE CODE APPENDIX

tests/logic.test.cjs · lines 1–27

මෙම කොටස ZIP source file එකෙන් ලබාගත් සැබෑ code එකකි. සම්පූර්ණ file එකට Project ZIP විවෘත කරන්න.

VISUAL STUDIO CODE — script.js

EXPLORER

```
Little-Star-Quiz
--- index.html
--- css/
  --- style.css
--- js/
  --- questions.js
  --- logic.js
  --- app.js
```

```
1  // Run with: node --test tests/logic.test.cjs
2  const test = require('node:test');
3  const assert = require('node:assert/strict');
4  const fs = require('node:fs');
5  const vm = require('node:vm');
6  const code = fs.readFileSync('js/logic.js', 'utf8');
7  const context = {};
8  vm.runInNewContext(code, context);
9  const { percentage, isCorrect, message, higherScore } = context.QuizLogic;
10
11 test('percentage rounds as expected', () => {
12   assert.equal(percentage(7, 10), 70);
13   assert.equal(percentage(2, 3), 67);
14 });
15 test('zero total is safe', () => assert.equal(percentage(0, 0), 0));
16 test('invalid numbers are safe', () => assert.equal(percentage(NaN, 3), 0));
17 test('correct answer is strict', () => {
18   assert.equal(isCorrect(1, 1), true);
19   assert.equal(isCorrect('1', 1), false);
20 });
21 test('top score message', () => assert.match(message(8, 10), /නියමයි/));
22 test('middle score message', () => assert.match(message(5, 10), /හොඳ/));
23 test('low score message', () => assert.match(message(1, 10), /කමක්/));
24 test('best score never decreases', () => {
25   assert.equal(higherScore(6, 9), 9);
26   assert.equal(higherScore(10, 9), 10);
27 });
```

CODE READING

මෙහි එක් function එකක්/element එකක් තෝරා එය කරන්නේ කුමක්දැයි ලියන්න.

REFERENCE / 2026

ඉගෙනීමට විශ්වාසදායක resources

මෙම පොතේ මූලික කරුණු සඳහා නිල සහ දැනට පවත්නා reference sources භාවිතා කර ඇත.

- MDN JavaScript Guide — developer.mozilla.org/en-US/docs/Web/JavaScript/Guide
- MDN JavaScript Curriculum — developer.mozilla.org/en-US/curriculum/core/javascript-fundamentals/
- MDN JavaScript Modules — developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules
- ECMA-262, 17th edition, June 2026 — 262.ecma-international.org/
- GitHub Pages Docs — docs.github.com/en/pages
- Source code: ZIP එකේ START_HERE_SINHALA.md සහ README.md

VISUAL LEARNING MAP



ACTIVITY

ඉගෙනගත් දේවල් සඳහා MDN Guide එකේ Functions section එක බලන්න.

FINISH

ඔබ දැන් JavaScript ඉගෙනගැනීමට පටන්ගෙන ඇත!

Code ලියන හැම දවසකම ඔබේ විශ්වාසය වැඩි වෙයි. එක project එකක් නිවැරදිව ඉගෙනගැනීම අලුත් projects ගණනක් ඉක්මනින් copy කිරීමට වඩා වටිනා දෙයක්.

- ඔබ ලියන code එක තේරුම්ගන්න.
- Code වෙනස් කර output එක බලන්න.
- Fail වූ පරීක්ෂණයක් තිබුණොත් fix කර නැවත run කරන්න.
- ඔබේ ඉගෙනීම ගුරුවරයෙකු හෝ මිතුරෙකු සමඟ බෙදාගන්න.

VISUAL LEARNING MAP



NOTE

Next mini project: JavaScript To-Do List එකක් ඔබම සැලසුම් කරන්න.