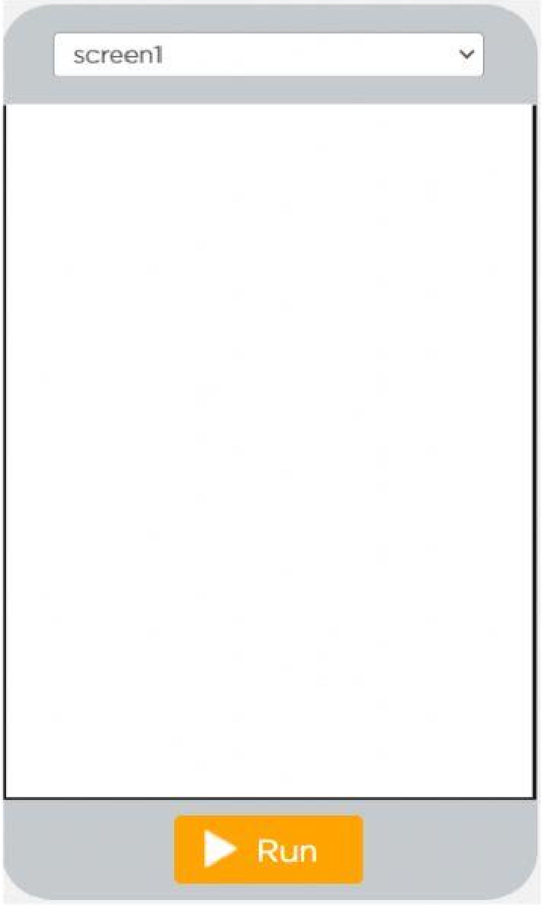
ඩී පී එඩියුකේෂන් කෝඩින් පාසල
කෝඩින් වෙසක් කලාපය - 2023

“HTML” ඇසුරින් ආලෝක රථා හදමු



DP EDUCATION Coding School

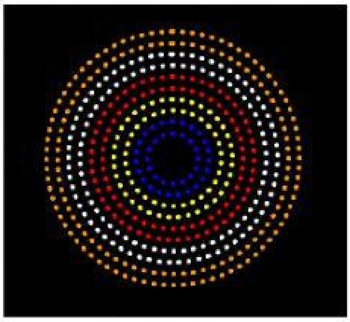
LIVEWORKSHEETS

මෙම ව්‍යාපෘතියෙන් ඉදිරිපත් කරන්නේ HTML ඇසුරින් සිදුකළ යුතු නිර්මාණයක් බැවින්, මේ වන විට ඩී පී කෝඩින් පාසලේ 249 ව්‍යාපෘතියේ සිට ඉදිරියට ඇති HTML අධ්‍යයනය ආරම්භ කර නැති අය පහත වීඩියෝව නරඹා glitch.com වෙබ් අඩවියේ SingUp වන ආකාරය දැනගන්න. එසේම 249 ව්‍යාපෘතියේ සිට ඉදිරියට ඇති ව්‍යාපෘති මගින් තවදුරටත් HTML පිළිබඳ කරුණු අධ්‍යයකය කළ හැකිය.

LIVEWORKSHEETS

HTML භාවිතයෙන් බුදුරැස් වළලු රටා නිර්මාණය කරමු.

- ❖ මේ සඳහා ඔබ විසින් HTML coding ඉගනගැනීමට භාවිතා කරන ලද “Glitch” platform එක භාවිතා කරමු. ඔබ විසින් new project එකක් ආරම්භ කර එය තුළ code කිරීම පටන්ගනිමු.
- ❖ බෞද්ධ කොඩියේ ඇති වර්ණ පහ දිස්වෙමින් නිවී පත්තු වන රටාව නිර්මාණය කරගන්නා අයුරු බලමු.



- ❖ පළමුව අවශ්‍ය වන වළලු නිර්මාණය කරගමු. මේ සඳහා එක් වර්ණයකින් වළලු දෙක බැගින් වළලු 10ක් නිර්මාණය කරගත යුතුය.
- ❖ මේ සඳහා html වල div tag භාවිතා කරමු. index.html file එකේ body එක තුළ div tags 10ක් code කරන්න.

```
<div id="thorana0"></div>
```

- ❖ මේ ආකාරයෙන් code කරන tag 10 යෙහි id එක “thorana0” සිට “thorana9” දක්වා ලබා දෙන්න.

- ❖ මෙසේ නිර්මාණය කරන ලද div tag 10 එක section එකක් ලෙස සලකා styles වෙනස් කිරීම සඳහා මේ සියල්ලම එක div tag එකක් තුළ wrap කරමු. එහි class name එක ලෙස “container” ලබා දෙන්න.

```
<body>
    <div class="container">
     <div id="thorana0"></div>
     <div id="thorana1"></div>
     <div id="thorana2"></div>
     <div id="thorana3"></div>
     <div id="thorana4"></div>
     <div id="thorana5"></div>
     <div id="thorana6"></div>
     <div id="thorana7"></div>
     <div id="thorana8"></div>
     <div id="thorana9"></div>
  </div>
</body>
```

- ❖ මේ ආකාරයෙන් නිර්මාණය කරගන්නා div tag එකක් සඳහා වෙබ් පිටුවේ දර්ශනය වන්නෙ කොටුවකි. දැන් අපි බලමු එය CSS භාවිතා කර ඉහත ආකාරයේ රවුමට හා බල්බ් සේ පෙනෙන ලෙස style කරගන්නා ආකාරය.
- ❖ මේ සඳහා style.css file එක තුළට ගොස් ඉහත div tag වල id එක සඳහා CSS නිර්මාණය කරමු.

LIVEWORKSHEETS

- පළමුව පසුබිම කළු පැහැගැන්වීම සඳහ body tag එකේ background-color – black කරමු. ඉන්පසුව අපේ සියළුම div tag අන්තර්ගත වන container div tag එක සඳහා styles නිර්මාණය කරමු. මෙමගින් div tag දහයේ position හා එකක් තුළ එකක් සේ පෙනීමට නිර්මාණය කරගත හැකිය. එහි styles පහත පරිදි වේ.

```
body{
 background-color: #000000;

}
```

```
.container{
  display: flex;
  align-items: center;
  justify-content: center;
  margin-top: 200px;
}
```

මෙහි margin-top මගින් container div tag එක ඉහළ සිට pixel 200 ක් පහළට කරයි.

- දැන් එක් එක් div tag එක සඳහා id එක භාවිතා කර ඒවා රවුම් හා බල්බ් ලෙස පෙනීමට styles නිර්මාණය කරමු.
  කුඩාම රවුම පළමු div tag එක වේ. එහි id එක “thorana0” වේ. ඒ සඳහා styles පහත පරිදි වේ.

```
#thorana0 {
  width: 50px;
  height: 50px;
  background-color: #00000000 ;
  position:absolute;
  border: #00000000 6px dotted;
  border-radius: 50%;
}
```

Border-color

මෙම රවුමේ background-color, border-color එක transparent කර ඇත. එනම් මෙය ඇති නමුත් ඇත්තටම නොපෙනේ. අප විසින් රටා නිර්මාණයට අවශ්‍ය JavaScript code මගින් මේවා දර්ශනය කිරීම සිදු කරයි. Background-color හා border-color සඳහා මෙම අගය ලබා දෙන්න #00000000.

- ඉහත කී පරිදී මෙම div tag සමචතුරස්‍රාකාර නිසා එය රවුමක් ලෙස පෙනීම සඳහා border-radius එක සඳහා 50%ක් ලබා දිය යුතුය.
- border එකෙහි ස්වභාවය dotted ලෙස ලබා දී එහි width එක 6px කිරීම මගින් ඔබට මෙම div tag එක බල්බ් වළල්ලක් ලෙස නිර්මාණය කරගත හැකිය.

```
border: #00000000 6px dotted;
```

- මේ ආකාරයටම අනෙකුත් div tag 9 සඳහාද ඒවාගේ id එක භාවිතයෙන් මෙම styles ඒවා සඳහාද ලබා දෙන්න. නමුත් එක් එක් වළල්ල අනෙකට වඩා විශාල නිසා width එක හා height එක 25 px බැගින් වැඩි කර ලබා දෙන්න.

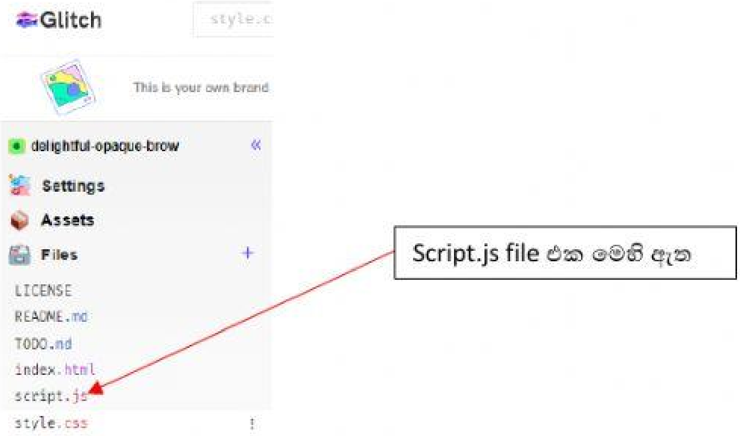
LIVEWORKSHEETS

```
#thorana1 {
  width: 75px;
  height: 75px;
  background-color: #00000000 ;
  position:absolute;
  border: #00000000 6px dotted;
  border-radius: 50%;
}
```

❖ දැන් අපි නිර්මාණය කර style කරගත් වළලු 10 රටාවක් සේ animate වීම නිර්මාණය කරගන්නා අකාරය බලමු.

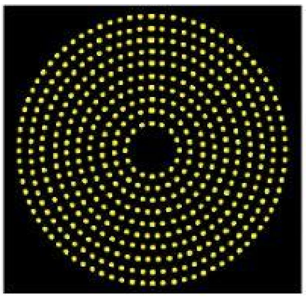
❖ මේ සඳහා ඔබට JavaScript භාවිතා කිරීමට සිදු වේ. පහත පැහැදිලි කර ඇති අකාරයට code කිරීමෙන් පහසුවෙන් ඔබට එය නිර්මාණය කරගත හැකි වනු ඇත. ඔබ App Lab එකේදී භාවිතා කරන ලද code මෙහිදී භාවිතා වන නිසා මෙම code ඔබට හුරු codes ඒවා වේ.

❖ JavaScript code කිරීම සඳහා script.js file එක භාවිතා කරන්නෙමු.

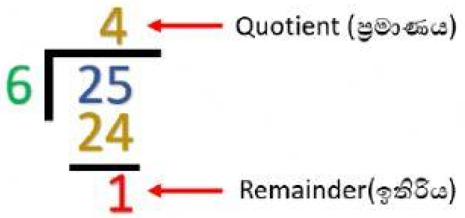


එය තුළ code කිරීම ආරම්භ කරමු.

❖ පළමුව වර්ණ පහක රටාව නිර්මාණය කිරීමට පෙර සියලුම වළලු එකකට පසු එකක් එකම ආලෝකයෙන් පත්තු වී සියල්ල නිවී යන රටාව නිර්මාණය කරමු.



❖ මෙහි Coding සඳහා Remainder එක ගැන දැනුම අවශ්‍ය වේ.



යම් සංඛ්‍යාවක් තවත් සංඛ්‍යාවකින් බෙදූ විට ඉතිරි වන අගය, Remainder(ඉතිරිය) වේ.

❖ Program වලදී Remainder ලබා ගැනීම ට “%” use කරනු ලැබේ. ඒ අනුව 25, 6 න් බෙදූ විට Remainder ලබා ගැනිම පහත පරිදි වේ.

```
var remainder;
remainder = 25 % 6;
```

මෙහිදී Remainder එකෙහි අගය 1 වේ.

❖ Remainder එක භාවිතා කර අපගේ රටාව නිර්මාණය කරගන්නා ආකාරය බලමු.

❖ පළමුව ඔබ count හා remainder ලෙස variable දෙකක් පහත පරිදී නිර්මාණය කරගන්න.

```
var count = 0;
var remainder ;
```

❖ App lab platform එකේදී භාවිතා කරන ලද timedloop code block එක වෙනුවට JavaScript වලදී අපි භාවිතා කරන්නෙ setInterval යන function එක වේ. එය පහත පරිදී වේ.

```
setInterval(()=>{
  //write here what do you want to do
},1000)
```

මෙමගින් 1000 ms වරක් යම් කිසි කාර්යයක් loop කරගත හැකි වේ. මීලි තත්පර 1000 වෙනස් කිරීම මගින් loop වීමේ වේගය වෙනස් කරගත හැකි වේ. (*//write here what do you want to do* මෙය comment එකක් වන අතර { } මෙම වරහන් දෙක තුළ ඔබට loop කිරීමට අවශ්‍ය වන code ඇතුළත් කරන්න.)

❖ setInterval තුළ පහත පරිදී code කරන්න.

```
setInterval(() => {
    remainder = count % 10;

    document.getElementById(`thorana${remainder}`).style.borderColor = "yellow";

    count = count + 1;

    if (remainder == 9) {
      setTimeout(()=>{
        for (var i = 0; i <= 9; i++) {
            document.getElementById(`thorana${i}`).style.borderColor = "#00000000";
        }
      },100)

    }
}, 150);
```

ඉහත code පැහැදිලි කරගමු.

❖ පළමුව අප ඉහතදී නිර්මාණය කරගත් “remainder” variable එකට අපගේ “count” variable එක 10 න් බෙදූ විට ඉතුරු වන අගය (remainder) සමාන කරගනී.

LIVEWORKSHEETS

මෙම අවස්ථාවේදී count variable එකෙහි අගය 0 නිසා (0 % 10 = 0) remainder එක 0 වේ. මක් නිසාද 0 ව 10න් බෙදූ විට ඉතිරිය 0ක් වන නිසා. එම නිසා “remainder” variable එකෙහි අගය 0 වේ.

- ❖ මෙහිදී අප “count” variable එක 10න් බෙදන්නේ අපගේ රටාවේ වළලු 10ක් ඇති බැවිනි
- ❖ JavaScript වලදී අපගේ index.html file එකෙහි ඇති element එකක් අල්ලාගැනීම සඳහා document.getElementById(“id”) භාවිතා කරයි. “id” සඳහා ඔබට අල්ලාගැනීමට අවශ්‍ය html element එකෙහි id එක ලබා දෙන්න.

```
document.getElementById("thorana0")
```

උදාහරණ ලෙස ඉහත code එක ගත් විට ඒ මගින් අප ඉහත නිර්මාණය කරන ලද කුඩාම වළල්ලට අදාල div tag element එක අල්ලගනී.

- ❖ මෙසේ element එකක් අල්ලාගත් පසු අපට JavaScript තුළදී එය වෙනස් කිරීමට හැකි වේ. ඒ සඳහා නව styles ලබා දිය හැකිය.
අපගේ රටාව නිර්මාණය සඳහා අප එය භාවිතා කරමු.

- ❖ ඉහත setInterval තුළ තිබූ `` document.getElementById(`thorana${remainder}`).style.borderColor = "yellow"; `` මෙම code කොටස මගින් ඒ ආකාරයෙන් අප html තුළදී div tag මගින් නිර්මාණය කරන ලද වළලු 10 යේ border-color එක “yellow” ලෙස වෙනස් කරයි.
- ❖ එහිදී id එක ලබා දී ඇති ආකාරය ඉතා වැදගත් වේ. එය මේ ආකාරයෙන් ලබා දෙන්න. `thorana${remainder}` . මෙම කොටස code කිරීමේදී ආරම්භක හා අවසාන symbol දෙක type කිරීම සඳහා keyboard එකෙහි ඇති පහත key එක භාවිතා කරන්න.

- ❖ මෙහිදී remainder variable එකේ අගය “thorana” යන වචනය සමග එකතු කර id එක නිර්මාණය කරයි. Remainder variable එකේ අගය 0 නම් id එක “thorana0” වේ.

- ❖ මේ නිසා remainder variable එකෙහි අගය වෙනස් වන විට ඊට අදාළ html element එක අල්ලාගන එහි border color එක වෙනස් කළ හැකි වේ.

- ❖ එය සිදු කිරීමෙන් අනතුරුව count variable එකෙහි අගය එකකින් වැඩි කිරීම සඳහා මෙම code කොටස භාවිතා කරන්න. `count = count + 1;`

LIVEWORKSHEETS

```
if (remainder == 9) {
  setTimeout(()=>{
    for (var i = 0; i <= 9; i++) {
        document.getElementById(`thorana${i}`).style.borderColor = "#00000000";
    }
  },100)

}
```

- ❖ ඉන්පසුව ඇති මෙම code කොටස මගින් එක් වටයක් සියළුම වළලු කහ වර්ණයෙන් දැල්වුන පසු ඒ සියල්ලගේම මිලි තත්පර 100 කට පසු නැවත border-color එක transparent කිරීම සිදු කරය.

- ❖ මෙහිදී remainder එක 9 වූ විට ඉහත code කොටස run වේ.

setInterval function එක මිලි තත්පර 150කට වරක් run වන නිසා count variable එකෙහි අගය 1 බැගින් සෑම වටයකදීම වැඩි වේ. මේ නිසා එක් එක් වටයේදී remainder එකෙහි අගය 0 සිට 9 දක්වා වෙනස් වේ. ඒ ආධාරයෙන් වළලු සියල්ලම දැල්වී තිවෙන ආකාරයට නිර්මාණය කරගත හැකිය.

- ❖ script.js file එක තුළ එක් වර්ණයකින් වළලු 10 දර්ශණය වීමට අදාළ සම්පූර්ණ code එක පහත පරිදි වේ.

```
var count = 0;
var remainder ;
setInterval(() => {
    remainder = count % 10;

    document.getElementById(`thorana${remainder}`).style.borderColor = "yellow";

    count = count + 1;

    if (remainder == 9) {
      setTimeout(()=>{
        for (var i = 0; i <= 9; i++) {
            document.getElementById(`thorana${i}`).style.borderColor = "#00000000";
        }
      },100)

    }
}, 150);
```

- ❖ දැන් ඉහත code එකම වෙනස් කිරීම මගින් වළලු දෙකකට එක් වර්ණය බැගින් වර්ණ පහකින් දර්ශනය වන රටාව නිර්මාණය කරමු.

- ❖ මේ සඳහාද ඉහත code සියල්ලම අවශ්‍ය වේ. අපි සිදු කරන්නේ remainder එකෙහි value එක වළලු පිහිටන අංකයට සමාන වේ නම් ඒවා එක් එක් වර්ණයෙන් පෙන්වීමයි.

- ❖ එය පහත පරිදි වේ.

LIVEWORKSHEETS

```
var count = 0;
setInterval(() => {
    var remainder = count % 10;
    if (remainder == 0 || remainder == 1) {
        document.getElementById(`thorana${remainder}`).style.borderColor = "blue";
    }
    else if (remainder == 2 || remainder == 3) {
        document.getElementById(`thorana${remainder}`).style.borderColor = "yellow";
    }
    else if (remainder == 4 || remainder == 5) {
        document.getElementById(`thorana${remainder}`).style.borderColor = "red";
    }
    else if (remainder == 6 || remainder == 7) {
        document.getElementById(`thorana${remainder}`).style.borderColor = "white";
    }
    else if (remainder == 8 || remainder == 9) {
        document.getElementById(`thorana${remainder}`).style.borderColor = "orange";
    }
    count = count + 1;

    if (remainder == 9) {
      setTimeout(()=>{
        for (var i = 0; i <= 9; i++) {
          document.getElementById(`thorana${i}`).style.borderColor = "#00000000";
        }
      },100)

    }
}, 150);
```

❖ මෙහිදී,

remainder එකෙහි අගය සමානයි 0 හෝ 1 නම් පළමු වළලු දෙක නිල් වර්ණයෙන්,

2 හෝ 3 නම් 3,4 වළලු දෙක කහ වර්ණයෙන්,

4 හෝ 5 නම් 4,5 වළලු දෙක රතු වර්ණයෙන්,

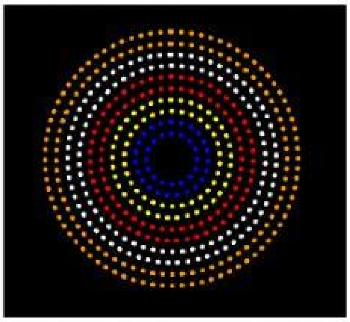
6 හෝ 7 නම් 6,7 වළලු දෙක සුදු වර්ණයෙන් හා

8 හෝ 9 නම් අවසන් වළලු දෙක තැඹිලි වර්ණයෙන් දර්ශනය වේ.

❖ Programming වලදී හෝ යන තර්කය ලබාගැනීමට “||” මෙම symbol එක භාවිතා කරයි.

ඉහත තනි වර්ණයෙන් රටාව නිර්මාණය කළ ආකාරයටම remainder එක 9ට සමාන වූ විට සියළුම වළලු වල border-color එක transparent කිරීමෙන් නිවී ගිය ආකාරය නිර්මාණය කරගත හැකිය.

❖ අවසානයේ ඔබට පහත ආකාරයේ රටාවක් නිර්මාණය කරගත හැකි වනු ඇත.



LIVEWORKSHEETS