Project 198
198 වන ව්‍යාපෘතිය

DP EDUCATION

Coding School

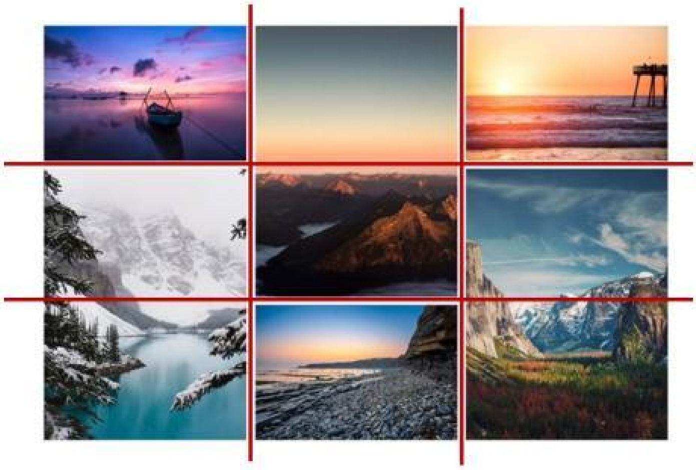
වෙබ් පිටුව බලන්න

Glitch ආරම්භ කරන්න

LIVEWORKSHEETS

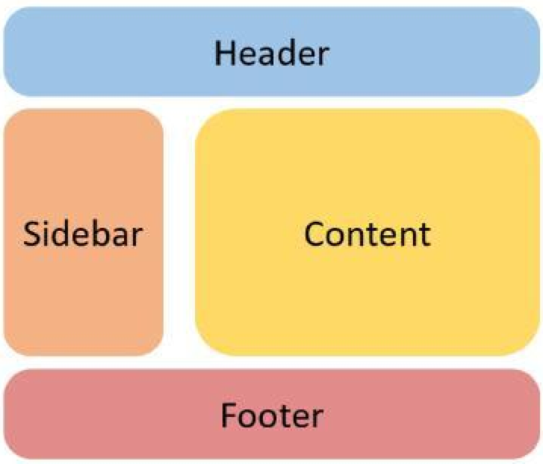
GRID

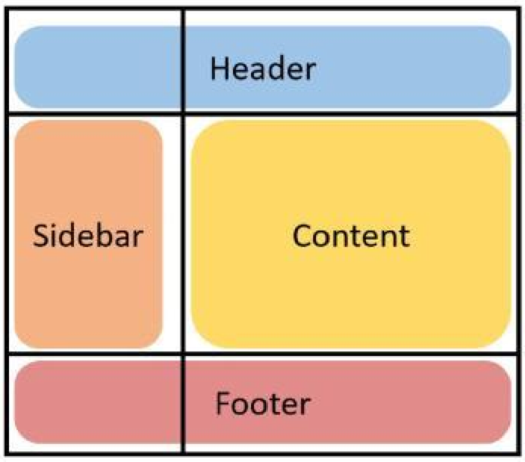
- සමහර වෙලවට content එකක layout එක rows තුල මෙන් ම columns තුලද පිහිටයි. එවිට අපි grids use කරනු ලැබේ.
- Grid එකක් use කරන ඉතා පොදු අවස්තාවක් නම් photo gallery එකක් නිමර්ාණය කිරීම.



මෙම photo gallery එක තුල එක එක size එකෙහි images තිබේ. නමුත් මෙම සියලු ම images පිහටා තිබේන්නේ grid එකක් තුලය. මේ photo gallery එක තුල row 3 ක් හා columns 3 ක් ඇත.

- Grids use කරන තවත් අවස්ථාවක් ලෙස බෝහො website වල layout එක ගැනිමට හැකිය.





LIVEWORKSHEETS

❖ දැන් අපි බලමු grid එකක් define කරන ආකාරය.

❖ මුලින් ම අපට අවශය වෙනවා container class එකක්. Container class එකෙහි display property එක grid ලෙස ලබා දිය යුතුය.

❖ ඉන්පසු අපට අවශ්‍ය rows ගණන සහා columns ගණන ලබා දිම සඳහා පහත ඇති properties දෙක use කරනු ලැබේ.

- grid-template-rows
- grid-template-columns

❖ දැන් index.html page එක පහත පරිදි code කරන්න.

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" href="style.css">
    <title>Grids</title>
  </head>
<body>
    <div class="container"></div>
  </body>
</html>
```

❖ දැන් style.css page එක පහත පරිදි code කරන්න.

මෙහිදි rows 3 ක හා columns 2 ක grid එකක් create කරනු ලැබේ.

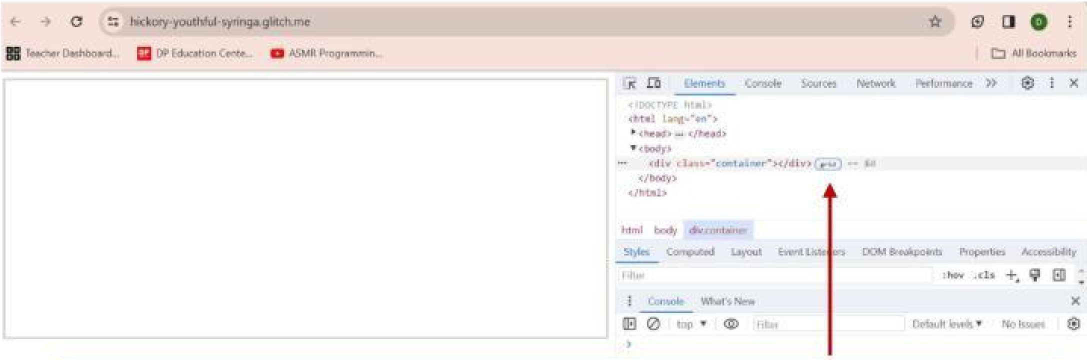
style.css   PRETTIER

```css
.container {
    display: grid;
    /* 3 x 2 */
    grid-template-rows: 100px 100px 100px;
    grid-template-columns: 100px 100px;
    border: 3px solid lightgray;
}
```

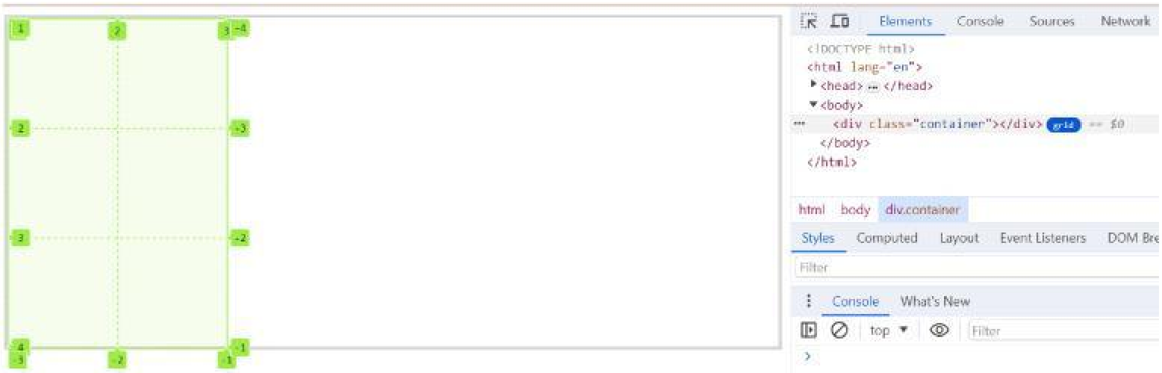
Rows 3 create කිරිම සඳහා මෙම property එක භාවිතා කරන අතර එක් එක් row එක සඳහා size එක මෙලෙස ලබා දෙනු ලැබේ.

Columns 2 create කිරිම සඳහා මෙම property එක භාවිතා කරන අතර එක් එක් Columns එක සඳහා size එක මෙලෙස ලබා දෙනු ලැබේ.

❖ දැන් preview කර බලන්න.



Webpage එක මත right click කර inspect මත click කරන්න. එවිට ඉහත ආකාරයට elements බලා ගත හැකිය. දැන් මෙහි ඇති grid මත click කරන්න. එවිට පහත ආකාරයට ඔබ නිර්මාණය කරන ලද grid එක view කර ගත හැකි වේ.



❖ ඉහත css එක පහත පරිදි shorthand ලෙස ලබා දිය හැකිය.

```css
.container {
    display: grid;
    /* 3 x 2 */
    grid-template-rows: repeat(3, 100px);
    grid-template-columns: repeat(2, 100px);
    border: 3px solid lightgray;
}
```

Repeat keyword එක යොදා ගනිමින් ඉහත ආකාරයට ලබා දිය හැකිය. මෙහිදි rows ගණන හෝ columns ගණන වරහණ තුල ඇති පලමු සංඛ්‍යාවන් ලබා දි ඇති අතර එහි size එක ඉන්පසුව ලබා දි ඇත.

LIVEWORKSHEETS

❖ දැන් පහත පරිදි code කරන්න.

```html
<div class="container">
    <div class="box">A</div>
    <div class="box">B</div>
    <div class="box">C</div>
    <div class="box">D</div>
</div>
```

```css
.container {
    display: grid;
    /* 3 x 2 */
    grid-template: repeat(3, 100px) / repeat(2, 100px);
    border: 3px solid lightgray;
}
.box {
    width: 5rem;
    height: 5rem;
    background-color: gold;
}
```

මෙයද repeat keyword එක භාවිතා කරමින් grid එකක් create කරන short ක්‍රමයකි. මෙහිදි / එකට පෙර ඇති කොටස මඟින් grid එකහි rows සඳහා ද / එකන් පසු කොටස columns සඳහා ද වේ.

❖ Create කරන ලද එක් එක් box පිලිවලින් එක් එක් columns තුල ඇත.

❖ Desktop සඳහා මෙලෙස view වුවද mobile view එක සඳහා මෙය සුදුසු නොවිය හැක. එම නිසා අපට අවශ්‍ය ආකාරයට මෙම box move කිරීමට සිදු වේ.

❖ ඒ සඳහා පහත ඇති properties භාවිතා කරනු ලැබේ.

- justify-items (along the **horizontal** axis)
- align-items (along the **vertical** axis)

LIVEWORKSHEETS

❖ දැන් පහත පරිදි code කරමු.

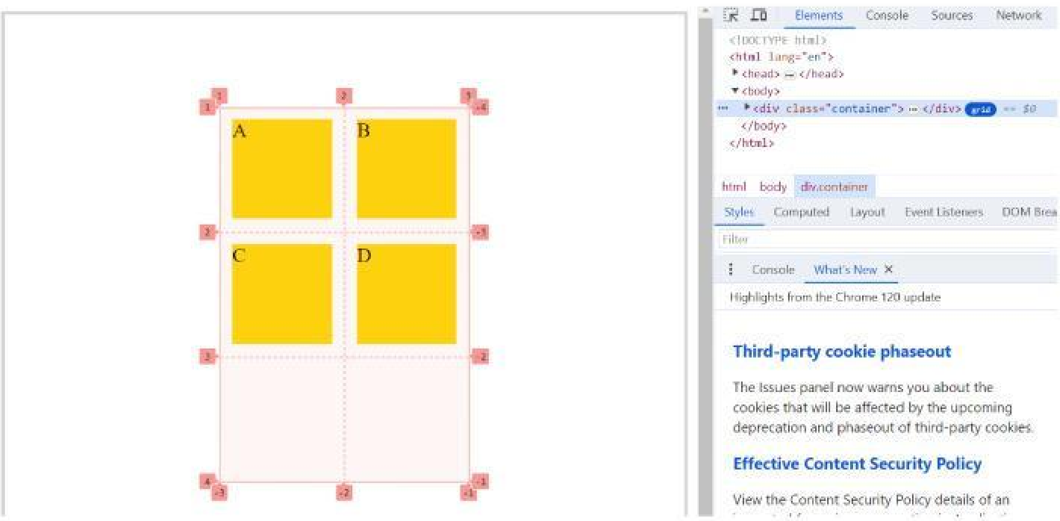
```css
.container {
    display: grid;
    /* 3 x 2 */
    grid-template: repeat(3, 100px) / repeat(2, 100px);
    border: 3px solid lightgray;
    justify-items: center;
    align-items: center;
    justify-content: center;
    align-content: center;
    height: 100vh;
}
```

මෙම මගින් grid ඒක තුල ඇති items grid ඒකේ තිරස් රේකාවට සපේක්ෂව center කරනු ලැබේ.

මෙම මගින් grid ඒක තුල ඇති items grid ඒකේ සිරස් රේකාවට සපේක්ෂව center කරනු ලැබේ.

මෙම මගින් grid එක, container එකේ තිරස් රේකාවට සපේක්ෂව center කරනු ලැබේ.

මෙම මගින් grid එක, container එකේ සිරස් රේකාවට සපේක්ෂව center කරනු ලැබේ.



මෙහිදි items මගින් grid එක තුල ඇති items සඳහා ද, content මගින් grid එක සඳහා ද style කරනු ලැබේ.

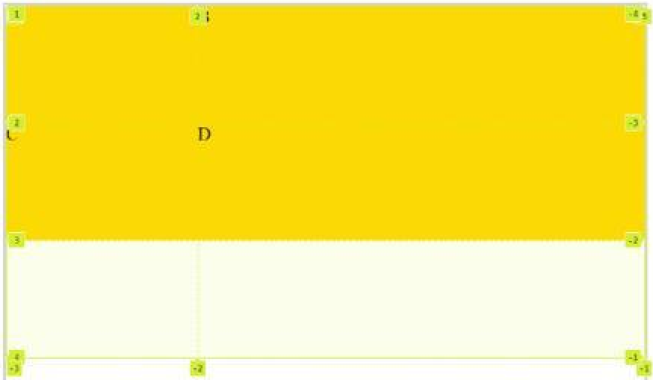
LIVEWORKSHEETS

❖ Rows හෝ columns වල size එක percentage එක මගින් ද ලබා දිය හැකිය.

❖ පහත පරිදි code කරන්න.

style.css   ✦ PRETTIER

```css
.container {
    display: grid;
    /* 3 x 2 */
    grid-template: repeat(3, 100px) / 30% 70%;
    border: 3px solid lightgray;
    height: 100vh;
}
.box {
    background-color: gold;
}
```



❖ දැන් columns එක සඳහා fix value එකක් සමඟ එක් column එකක් හා අනෙක් columns 2 ක percentage එක මගින් ලබා දී බලමු.

```css
.container {
    display: grid;
    /* 3 x 2 */
    grid-template: repeat(3, 100px) / 25px 30% 70%;
    border: 3px solid lightgray;
    height: 100vh;
}
.box {
    background-color: gold;
}
```

එවිට 25px fix column එකක් ද, percentage එක මගින් ලබා දුන් columns 2 ක ඉතිරි width සඳහා සකස් නොවී මුලු width එකම ලබා ගෙන ඇත.

❖ එසේ නොවී ඉතිරි width එක පමණක් use කිරීම සඳහා පහත ආකාරයට values ලබා දිය හැකිය.

style.css   ✦ PRETTIER

```css
.container {
    display: grid;
    /* 3 x 2 */
    grid-template: repeat(3, 100px) / 25px 30fr 70fr;
    border: 3px solid lightgray;
    height: 100vh;
}
.box {
    background-color: gold;
}
```

LIVEWORKSHEETS

- Grid හි column අතර හා row අතර හිඩසක් තැබීමට පහත properties භාවිතා කරනු ලැබේ.
  - row-gap :- මෙමගින් rows අතර හිඩසක් තැබීම සිදු කරයි.
  - column-gap :- මෙමගින් columns අතර හිඩසක් තැබීම සිදු කරයි.
  - gap :- මෙමගින් rows හා columns අතර හිඩසක් තැබීම සිදු කරයි.

- Grid එක තුල ඇති items වල පිහිටීම වෙනස් කිරීම සඳහා පහත පහත properties භාවිතා කරනු ලැබේ.
  - grid-row
  - grid-column
  - grid-area

Columns හා rows අතරේ ඇති හිඩසේ value එක සමාන විට පහත ආකාරයට එම value එක ලබා දිය හැක. එසේ නොවන විට මුලින් ම rows අතරේ ඇති හිඩසේ value එක ද columns අතරේ ඇති හිඩසේ value එක ලබා දිය යුතුය.

Example:

```
gap: 20px 10px;
```

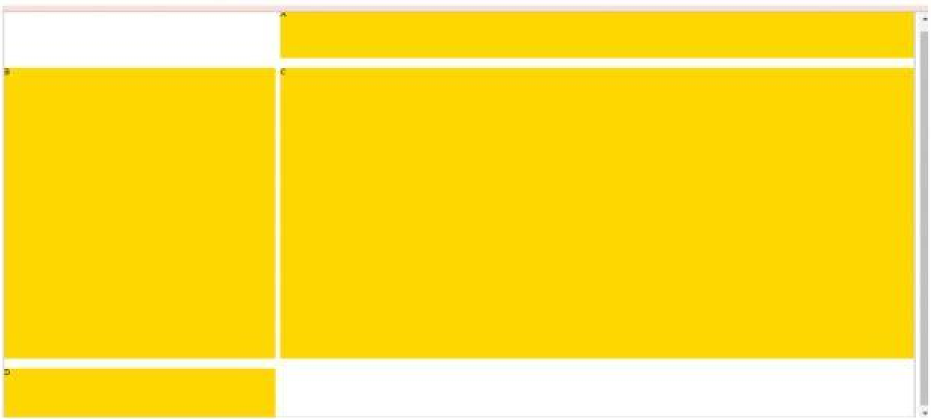
- දැන් පහත පරිදි code කරමු.

```html
<div class="container">
    <div class="box box-one">A</div>
    <div class="box">B</div>
    <div class="box">C</div>
    <div class="box">D</div>
</div>
```

```css
.container {
    display: grid;
    /* 3 x 2 */
    grid-template: 100px auto 100px / 30fr 70fr;
    gap: 10px;
    border: 3px solid lightgray;
    height: 100vh;
}
.box {
    background-color: gold;
}

.box-one {
  grid-column: 2;
}
```

Class එක box-one වන box එක grid එකහි දෙවන column එක සඳහා ලබා දී ඇත.



LIVEWORKSHEETS

❖ දැන් පහත පරිදි code කරමු.

```html
<div class="container">
    <div class="box box-one">A</div>
    <div class="box">B</div>
    <div class="box">C</div>
    <div class="box box-four">D</div>
</div>
```

style.css   PRETTIER

```css
.container {
    display: grid;
    /* 3 x 2 */
    grid-template: 100px auto 100px / 30fr 70fr;
    grid-template-areas:
      "header  header"
      "sidebar main"
      ".       footer";
    gap: 20px 10px;
    border: 3px solid lightgray;
    height: 100vh;
}
.box {
    background-color: gold;
}

.box-one {
  grid-area: header;
}

.box-four{
  grid-area: footer;
}
```

Grid එක තුල ඇති items වල පිහිටීම සැකසීම සඳහා මෙම property ඒක භාවිතා කරයි.

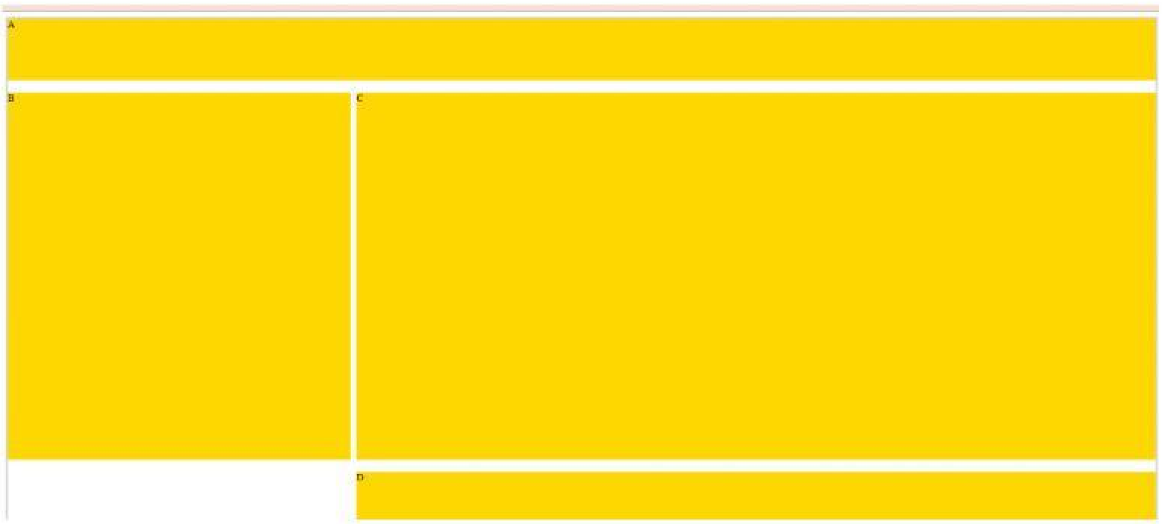
මෙහි පහලට ඇති පේලි මගින් එක් එක් row ද එම පේලියේ ඇති වචන මගින් එක් එක් columns ද පෙන්වයි.

පලමු හා දෙවන columns සඳහා Class එක box-one වන box එක ලබා ගැනිම සඳහා header ලෙස ලබා දී ඇත.

තුනවන column සඳහා box ඒක ලබා ගැනිම සඳහා sidebar ලබා දී ඇත. හිඩසක් සහිතව box එකක් ලබා ගැනිමට ඉන් පසු main ලෙස ලබා දෙයි.

පස්වන column එක හිසව තැබිම සඳහා . නබා ඇත.

ඉන්පසු හයවන column එක සඳහා footer ලෙස ලබා දෙයි.



LIVEWORKSHEETS