Part D-Diary

```
/* Nth Pesudo Class */
/* The CSS rule li:nth-of-type(2) targets the second <li> (list item) in a list and applies a set
of styles to it. */
li:nth-of-type(2){
   background-color:  #87d684;
   border: solid 2px rgb(0, 0, 0);
   border-color:rgb(207, 207, 142);
   border-radius: 6px;
   color: rgb(0, 0, 0);
}

/* Validity Pseudo class */

/* The CSS rule .NewEmail:invalid targets elements with the class NewEmail that are in an
invalid state. Typically, this rule is used with form elements,
such as an input with the NewEmail class that fails validation.  */

.NewEmail:invalid{
   background-color: rgb(140, 200, 135);
   border: 2px solid rgb(179, 90, 90);
}

/*Negation Pseudo Class*/

The CSS rule td:first-of-type:not(colspan)::first-letter targets the first letter of the content in
the first <td> (table cell) of each row where the first <td> is not a colspan (colspan attribute is
not applied).


td:first-of-type:not(colspan)::first-letter{
   background-color:lightblue;
   font-size:20px;
   color: black;
   font-weight: bold
}

td:first-of-type:not(colspan)::first-letter{
      font-size: 17px;
   }

/*Borders*/

/*The border property you've provided, border: solid gray;, sets a solid gray border for an
element.*/
border: [border-width] [border-style] [border-color];
```

```
table{
    margin: auto;
    border:solid gray;
    width: 96%;
    height:180px;
}

.txt-overflow{
    overflow: hidden;
    white-space: nowrap;
    width: 130px;
    overflow: hidden;
    border: 1px solid #000000;
    color: #e3b5b5;
    font-size: 19px;
    font-family: 'Times New Roman', Times, serif;
}

.NewEmail:invalid{
    background-color: rgb(140, 200, 135);
    border: 2px solid rgb(179, 90, 90);
}

li:nth-of-type(2){
    background-color:  #87d684;
    border: solid 2px rgb(0, 0, 0);
    border-color:rgb(207, 207, 142);
    border-radius: 6px;
    color: rgb(0, 0, 0);
}

/*Rounded Corners*/

/*To create rounded corners in CSS, you can use the border-radius property. This property
allows you to define the radius of each corner of an element*/

.button{
    /* background-color: #6f1e37; */
    border-radius: 6px;

.about-col-1 img{
    border-radius: 15px;
}
.services-list div{
    background-color: transparent;
    border-radius: 20px;
}
.btn2{
    border-radius: 6px;
```

```
}

form input, form textarea{
   border-radius: 3px;
}

.blog-img{
   border-radius: 10px;
}

.services-list-1 div{
   border-radius: 20px;
}

.navbar a{
   border-radius: 4px;
}
li:nth-of-type(2){
   border-radius: 6px;
}

/*Gradient*/

/*background-image: linear-gradient(red, rgb(108, 144, 108));: Sets a linear gradient as the background image. The gradient transitions from red to an RGB color (in this case, a shade of green: RGB (108, 144, 108)).*/
Linear Gradient: A linear gradient transitions colors along a straight line.
Radial Gradient: A radial gradient transitions colors outward from a central point.


#grad1{
   background-color: red; /* For browsers that do not support gradients */
   background-image: linear-gradient(red, rgb(108, 144, 108));
}
p:empty{
   background-color: red; /* For browsers that do not support gradients */
   background-image: linear-gradient(red, rgb(108, 144, 108));
}

.p1{
   background-color: rgb(233, 209, 209); /* For browsers that do not support gradients */
   background-image: radial-gradient(rgb(238, 212, 212), rgb(190, 226, 190));

}

/*Transforms*/

/*transform: translateY(-5px);:
Applies a transformation that moves the element 5 pixels upward along the Y-axis when hovered over. This creates a visual effect of the button shifting upward*/
```

```
.button:hover:hover{
   transform: translateY(-5px);
}

.header-text h1 span{
   transform: translateX(-5px);
}

.services-list div:hover{
   transform: translateY(-10px);
}

.services-list-1 div:hover{
   transform: translateY(-10px);
}

/*The @keyframes rule you provided defines a keyframe animation named slidedouble. This
animation uses the transform property to create a vertical translation effect.*/

@keyframes slidedouble{
0%{
   transform: translateY(0px);
}
50%{
   transform: translateY(-10px);
}
100%{
   transform: translateY(0px);
}
}

.button:hover:hover{
   transform: translateY(-5px);
}

/*Transitions*/

/*transition: all .3s;:  Applies a transition to all properties with a duration of 0.3 seconds*/
/*transition-delay: 2s;: This property adds a delay of 2 seconds before the transition starts
when the hover state is triggered. It introduces a delay before the overflow property takes
effect.*/
Get all the transitions here

.txt-overflow:hover{
   overflow: visible;
   transition-delay: 2s;
}

/*Animations*/
```

/*The @keyframes rule you provided defines a keyframe animation named slidedouble. This animation uses the transform property to create a vertical translation effect*/

/*In CSS, @keyframes is a rule that defines a set of styles or transformations at specific points in an animation. It is used in conjunction with the animation property to create smooth and controlled animations on HTML elements.*/

```
@keyframes slidedouble{
0%{
   transform: translateY(0px);
}
50%{
   transform: translateY(-10px);
}
100%{
   transform: translateY(0px);
}
}


.services-list-1 h2{
   animation-name: slidedouble;
   animation-duration: 3s;
   animation-timing-function: ease;
   animation-iteration-count: infinite;
   transition: all .3s;
}

.paragraph1::first-letter{
   animation-name: slidedouble;
   animation-duration: 3s;
   animation-timing-function: ease;
   animation-iteration-count: infinite;
   transition: all .3s;
}
```

@FontFace

-Defines a custom for a webpage
Font Family -This prioritized list of font families for selected elements. This will check the font family and if It's not available it will try out the next one.

Src - This specifies the source of the font file. This is a URL. To work this correctly you need to give the actual font file name path.

Google Fonts

Get the link for the html file then access the CSS file by giving it's font-family name for the font-family attribute.

Colours

Color: #ffffff

Semi-Transparent Color
-In the heading, I have used a semi-transparent color inside the media queries. When I make the website responsive it will change it into rgba(255,555,255,0.5) with 50% opacity and 50% Transparent. 0 is fully transparent and 1 is fully Opaque.

Shadows

Text Shadow-In the "services-list div h2& services-list-1 div h2" I have made all the topics in text shadows. It appears everywhere in h2 headings under the services list.

The text-shadow property is used to apply a shadow to the text element.
Ex-text-shadow-2px 2px 2px #fffff
1. Horizontal offset
2. Vertical Offset
3. Blur Radius
4. Color

Box-Shadow

1. box shadow:2px 1px
    - 2px- Horizontal offset(2px to right)
    - 1px- Vertical Offset(1px downwards)

2.box shadow:1px 1px 20px #ffffff
    - 1px- Horizontal offset(1px to right)
    - 1px- Vertical Offset(1px downwards)
    - 20px-blur radius
    - #fffff-Colour

3.box shadow:rgba(33,35,38,0.1) 0px 10px 10px -10px
- Colour
- 1px- Horizontal offset(1px to right)
- 1px- Vertical Offset(1px downwards)
- 20px-blur radius
- -10px -Spread Radius

Text Effects

Text-Overflow:
    Overflow: hidden

This property is used to hide any content that overflows the specified width and height. When it gets hovered it will allow the content to overflow the specified width to become visible.

Project div p- If it's overflows the specified width and height the text will be hidden.

Attribute Selector

a[target]
a→select anchor tags.
Select those anchor tags that have a target attribute. Regardless of its value.

a[target="_top"]
a→Select anchor elements
select those anchor elements that have a target attribute with the exact value of "_top"

Pseudo Elements with Generated Content

:: first-letter- targets the first letter of the text content within the text content within an element with the class "para-cert-1". This allows you to apply specific styles.

::before-The css rule "address ::before" with the content property is used to insert content before the content of an <address>.

:: after

:: Placeholder-The CSS rule ".contant -right-form::placeholder" targets the placeholder text inside input elements that are descendants of the form.

User action pseudo-class

: hover

Structural Selectors

: root→The root pseudo class in CSS is used to define global css variables. These variables are also known as customer properties. This can use throughout your style sheet. Once root is defined you can use them in other parts of your CSS.

: empty→The css rule target "p:empty" the paragraph element(paragraph with no elements) and applies to them.

: -nth-of-type

- This rule targets the second <li> list item and applies a set of styles.

Negation Pseudo Class

not(s)→(td:first-of-type):[not(colspan)]::first-letter, targets the first letter of the content in the first <td>

- Select the first <td> element in each of the table
- If it's a colspan td this rule won't be applied.

Nth Pseudo Class

-nth-of-type

- This rule targets the second <li> list item and applies a set of styles.

Relational Selector

/*The styles would be applied to elements with the "class .project-div and .project-div-1" only if they contain an img element as a descendant.*/

```
.project-div:has(img){
   margin-left: -10px;
   border-radius: 10px;
   border: 5px double #730058f7;
}
.project-div-1:has(img){
   margin-left: -10px;
   border-radius: 10px;
   border: 5px double #730058f7;
}

ul li{

}

nav ul li a {

}

nav ul li a::after {

}

nav ul li a:hover{

}
```

Child Combinator

/*In this case, the CSS rule a > i would apply to the <i> element inside the anchor (<a>) because it is a direct child.*/

```
a >i{
    color: #beb7ff;
}
```

Week 1 october 11th

① In html5 why we should use html5 when designing explain benefit?

existing features - New Elements

* one benefit of using html5 when designing is its 'Native support for multimedia elements'. Html5 introduces new tags like <audio>, and <video> which allows developer to embed audio and video content directly into web pages without relying on third party plugins like 'flash'.

② scripting based - API

* The benefit of scripting based API in html5 is that they allow the developers to create 'dynamic and interactive web pages'. with scripting based APIs such as Javascript API in html5, developers can manipulate and update content on a web page in real time based on user action and other events.

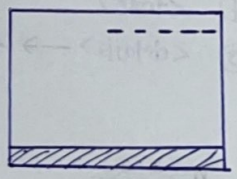
Benefit. <drag and drop functionality>.
<microdata>

Benefits of CSS3

* New features in CSS3 include support for additional selectors, drop shadows, rounded corners, multiple backgrounds, animations, transparency and much more.

These tags No need closing tags.

 <img>
<input> <link>.



semantic elements

* includes several semantic elements to help organise documents
- , , <nav>

No meaning semantic

<header> october 18th

* The <header> element represents a 'group of introductory or navigational aids'. The <header> element 'defines a page area that typically contains a logo, title and a navigation bar.' The header can also be used inside the semantic elements such as <article> or <section>.

<section> --> section header might be containing the section heading and author name.

<article>, <section>, <aside> can have their own headers.

<section> ---> The section element represents a generic section of a document or application. A section in this context is a grouping of content with a heading.

↑ Semantic meaningful.

<article> ---> similar to the <section> element. The article element represents a self contained composition in a document that is 'independently distributable or reusable'.

<nav> --> * Represents a group of navigation links.
* Represents a section of a page that links to other pages or to parts within other different pages

<aside> --> * An aside element is appropriate when it is used to represent 'content that is not the primary focus' of an article or page, but it is still related to the article or page.
* ලිපියක හෝ පිටුවක ප්‍රධාන අවධානය යොමු නොවන නමුත් එය කෙසේ හෝ ලිපිය හෝ පිටුව සම්බන්ධ අන්තර්ගතය නිරූපණය කිරීමට භාවිතා කරන මූලකයක් වේ.

aside and article.

① What's the difference between ~~footer and header~~?

<aside> - The element is used to define content that is tangentially related to the content around it. It typically represents content that is considered as a side bar, related links or advertisments.
* This can be placed inside a <article> or <section> element.

<article> - The <article> element is used to represent a complete, independent piece of content that can be reused. The content make the sense its own.

<<footer>>

* The footer element represents a footer for its nearest 'sectioning content or sectioning root element'
* Much like aside and header a footer element is not defined in terms of its position on the page. Hence it does not have to appear at the end of the section or at the bottom of the page. Most likely it will but it's not required.

october 29th

<progress> , <meter>

<progress max="100" value="70">70%</progress>.
<meter value="70" max="100" min="0" title="GB">400GB</meter>

④ Briefly explain about the date time in html5.
* The time element also allows you to express dates and times in whichever format you like. It will return the date and time value in the datetime attribute. This value can be converted into localized or preffered form using Javascript.

ex: <time datetime=""> July 7 </time>. .

<time datetime="2022-10-12T 6:24:34.042"> 12 october </time>
* Include the time along with the date.
↑ The 'T' character is used to start of the time
format is should be like this.

Week 2 — November 1st

ul li - descendant selector.
ol li

ol > li child selector matches <li> in <ol> but not nested in

UI pseudo class
: enabled
: disabled
: checked.
: indeterminate.

Form related of pseudo class
: default
: valid
: invalid
: required
: optional
: in range
: out-of-range.
: read-only.
: read-write.

colours
Transparent - rgba(0,0,0,0)
semi Transparent - rgba(0,0,0,0.5)

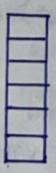
Gradients - linear Gradient();
radial Gradient();
conic Gradient();
repeating- " "();

Week 9 - Responsive — November 8th

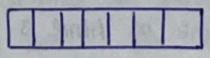
* A design that responds to the needs of the users and the devices they are using. The layouts changes based on the size and capabilities of the device.

* A webpage looks good in all devices.

Display : flexbox →
* one dimention.
* Great space distribution of items on the same axis



Display : flex



Display : grid
* two dimention
* Great layout that
reaare more contro
with rows and colum

To make it Responsive

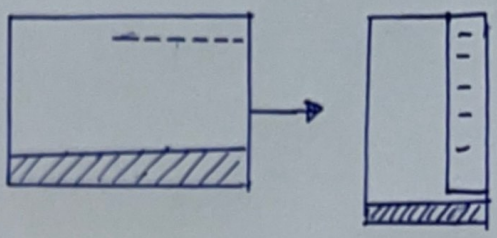
① set the viewpoint
<meta name = "viewpoint" content = "width = device-width , intial-scale = 1.0">

This sets the viewpoint of the page, and gives the browser instructions on how to control the page dimensions and scaling.

Sets the width of the page to follow the screen-width of the device.

sets the initial zoom level when the page is first loaded by the brow

(2) set the image max-width

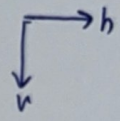


* Images larger than the viewpoint will cause horizontal scrolling.
* common way to deal with this problem is make your images max width into 100%.

```
img {
    max-width : 100%
    display : block
}
```

(3) set the Layout

* Modern css gives the layout techniques by creating flexible grids.
  - flexbox
  - Grid-
  - Multicol.



Media Queries

* Media queries are simple filters that can be applied to css styles. They make it easy to change styles based on the types of device rendering the content, or the features of the device. specially we check these attributes inside a media query. (1) width
  (2) height
  (3) orientation
  (4) ability to hover
  (5) used to touchscreen

media queries based on viewpoint

* Media queries makes to create a responsive experience where specific styles are applied in small screens, large screens and between screens. The feature is detecting is 'screen size'. It is based on below attributes
  (1) width.
  (2) height
  (3) orientation
  (4) aspect-ratio.

```
@media (max-width : 600px) {
    styles for 600px and down ↓ 600px
}
@media ( min-width : 610px) {
    styles for 601 px and up ↑ 601px
}
```