DAY1 – HTML

Box를 잘 나눠야 labelling이 쉬움

태그 – BOX(사용자에게 보여지지 않음), ITEM

|  |  |
| --- | --- |
| Header | |
| Nav | |
| Aside | Main  (Sections>Articles) |
| footer | |

Box: header, footer, section, article, div(묶어서 스타일링), nav, aside, span, main, form

Item: a, button, input, label, img, video, audio, map, canvas, table

블록레벨인가 인라인레벨인가? Block/Inline

블록레벨의 요소는 한 줄에 하나만 들어가게끔

인라인 레벨 태그는 공간이 허용하는 한 옆에 다른 것도 배치가 가능하다

태그와 엘리먼트

태그의 시작과 끝=엘리먼트 <p class=”editor-note”>My cat is very grumpy</p>

태그 안에 들어가는 내용=컨텐츠 <p class=”editor-note”>My cat is very grumpy</p>

태그 안에 들어가는 내용=어트리뷰트 <p class=”editor-note”>My cat is very grumpy</p>

<b></b>태그는 인라인 태그임

박스 찾기-F12 or ctrl shift I (element 익스패터 tool)

|  |  |
| --- | --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width">  <title>JS Bin</title>  </head>  <body>  <!-- a -->  <a href="https://google.com" target=\_blank>Click</a>  <p>This is a sentence. <b>That</b> is..</p>  <p>This is a sentence. <span>That</span> is..</p>  <p>This is a sentence. <div>That</div> is  <!-- ol, ul, li -->  <ol type="i" reversed> <!-- ol>li\*3>tab-->  <li>1</li>  <li>2</li>  <li>3</li>  </ol>  <ul>  <li>Hello</li>  <li>Hi</li>  <li>hehe</li>  </ul>  <!-- input -->  <label for="">Name: </label>  <input id="input\_\_name" type="text"></input>    </body>  </html> |  |

<CSS>

잘 정리한 HTML 페이지를 스타일리쉬하게 꾸며주는 것

1. 의미, 정의

Cascading Style Sheet (Author>User>Browser)

Author style: css, 우리가 지정한 것

User Style: 사용자가 브라우저에서 스타일링을 바꾸는 것

Browser: 브라우저 상에서 기본적으로 지정된 스타일

Cascading의 연결고리를 끊어내는 것 = ! important (뭔가 잘못했을 때,,, 가능하면 X)

1. 선택자

Universal \*

Type Tag div

ID #id

Class .class

State :

Attribute []

|  |  |
| --- | --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <meta name="viewport" content="width=device-width">  <title>JS Bin</title>  </head>  <body>  <ol>  <li id="special">First</li>  <li>Second</li>  </ol>  <h1 id="special">Hello</h1>  <button>Button 1</button>  <button>Button 2</button>  <div class="red"></div>  <div class="blue"</div>  <a href="naver.com">Naver</a>  <a href="googlenaver.com">Google</a>  <a>Empty</a>  </body>  </html> | /\*selector{  property: value;    } \*/  \*{  color: green;  }  li {  color: blue;  }  li#special{  color: pink;  }  .red {  width: 300px;  height: 30px;  background: yellow;  }  button:hover {  color: red;  background: beige;  }  a[href$=".com"] {  color: purple;  } |

|  |  |
| --- | --- |
| .red {  width: 100px;  height: 100px;  padding: 20px 0px;  margin: 20px;  border: 2px dashed red;  /\*border-width: 2px;  border-style: solid;  border-color: pink;  \*/  background: yellow;  } |  |

1. 스타일링
2. 헷갈리는 컨셉
3. CSS 꽃: Flex box

DAY2

텍스트이(가) 표시된 사진

자동 생성된 설명

\*position

기본값-static (html에 정해진 순서대로 브라우저 상에 표기)

Position: relative로 바꾸면 정한대로 바꿈 (원래 자리에서 상대적으로)

Absolute-내 아이템이 담겨있는 박스 안에서 위치를 바꿈

Fixed- 상자 안에서 벗어나서 윈도우에서, 웹페이지 안에서 움직여 간 것

Sticky- 원래 있던 자리에 있으면서 스크롤링 되어도 없어지지 않고 고대로 붙어있음

Block은 한줄에 하나만 위치할 수 있다.

inline-block은 한줄에 여러개 위치할 수 있다.

inline은 안에 존재하는 콘텐츠의 양에 따라 그 크기가 가변한다.

relative는 원래 있어야 하는 위치에서 상대적으로 이동한다 !!!

absolute는 아이템이 담겨있는 상자 안에서 위치가 변한다 !!!

fixed는 상자를 벗어나!! 윈도우 내부에서 위치가 변한다 !!!

position 정리

static : 기본 위치

relative : static 위치에서 이동

absolute : 바로 상위 박스위치에서 이동

fixed : 페이지상에서 이동

sticky : static위치 계속 유지

\*Flexbox

Float: left, center, right (이미지와 텍스트를 어떻게 배치할 것인가)

Left- 이미지는 왼쪽, 텍스트는 이미지를 감싸게

1.

Container- display, flex-direction, flex-wrap, flex-flow, justify-content, align-items, align-content

Item- order, flex-grow, flex-shrink, flex, align-self

2. 중심축과 반대축

Main axis, cross axis (왼 -> 오 정렬? 수직 축이 반대축)

Flex-direction // 어느방향으로 아이템을 정렬할것인가

Flex-rap: 화면이 좁아졌다 넓어질 때 아이템을 줄일 것인가 다음줄로 넘길 것인가

== flex-flow: column nowrap;

Justify // 가로축으로 아이템들을 어떻게 배열 // main axis

Align-items // 수직으로 아이템 어디다가 배치 baseline, center …

Flex-grow // 반응형을 만들기 위해 중요한 것, 컨테이너가 커질 때 꽉 채우려고 아이템의 크기가 커짐

Flex-shrink // 아이템들이 점점 줄으면 어떻게 작아져야 하는지 지정

Flex-basis // 기본값 auto, grow나 shrink를 지정하지 않고 %를 지정하면 알아서 됨

Align-self // 아이템별로 아이템을 정렬할 수 있음

*<!--emmet-->*

*<!-- !+ tab -> HTMl basic start-->*

*<!-- .class -> div class, #id -> div id-->*

<div> *<!--div>ul>li\*3-->*

  <ul>

    <li></li>

    <li></li>

    <li></li>

  </ul>

</div>

<div> *<!--div>ul+ol-->*

  <ul></ul>

  <ol></ol>

</div>

*<!--^-->*

<div> *<!-- div>ul>li^ol-->*

  <ul>

    <li></li>

  </ul>

  <ol></ol>

</div>

*<!--() group-->*

<div> *<!--div>(header>ul>li\*2>a)+footer>p-->*

  <header>

    <ul>

      <li><a *href*=""></a></li>

      <li><a *href*=""></a></li>

    </ul>

  </header>

  <footer>

    <p></p>

  </footer>

</div>

*<!--p{Hello}-->*

<p>Hello</p>

*<!-- {$}-->*

<p *class*="class1">item 1</p>

<p *class*="class2">item 2</p>

<p *class*="class3">item 3</p>

<p *class*="class4">item 4</p>

<p *class*="class5">item 5</p>

*<!--p lorem-->*

<p>Lorem ipsum dolor sit.</p>

DAY3

반응형 웹, 웹 개발할 때 알아야 하는 Responsive Web Design

큰 화면, 작은 디바이스에 맞춰서 알아서 반응되게 웹사이트를 만들어줘야 함

컨텐츠를 물과 같이 유동적으로 만들어라

반응형으로 잘 만들어진 사이즈 – 깃허브

Fluid layout으로 만들기, 고정된 px로 만들지 말고 flex grid, box, %, vw, vh 만들기

어느정도 포인트에 가면 레이아웃이 재배치 – Media Queries

Mobile-320~480

Tablet-768-1024

Desktop-1024~

@media screen and (min-width: 800px) {

.container{

Width: 50px;

}

}

웹사이트에 들어가는 아이콘들 Font Awesome 사용