**Web2 – css**

**css 사용 이유**

정보와 디자인의 분리

효율성 (중복의 제거로 유지, 보수, 가독성 장점)

**html 태그**

1. <!—

무시할 내용

-->

주석 태그

1. <style> </style>

<style>

···

</style>

···의 내용을 css 문법으로 해석하도록 하는 태그

이 뒤로 나오는 css 문법은 전부 ···에 들어가는 내용

1. style 속성

style="color:red"

1. property 검색
2. selector

class를 지정해 원하는 그룹을 selector로 사용, class="saw"

<li><a href = "1.html" class="saw">HTML</a></li>

<li><a href = "2.html" class="saw">CSS</a></li>

style 태그 안에

.saw {

color:gray;

}

만약 두개의 class를 두고 싶으면 아래처럼 띄어쓰기로 구분

style 태그 안에 늦게 등장하는 class 속성 영향을 받음 (가까이 있는 명령을 받음)

<li><a href = "2.html" class="saw active">CSS</a></li>

따라서 좋은 방법X

∴ class 선택자보다 우선인 id 선택자 사용, id="active" (한번만 사용 가능 – 중복 불가)

<li><a href = "2.html" class="saw" id="active">CSS</a></li>

style 태그 안에

#active {

color:red;

}

tag 선택자 < class 선택자 < id 선택자

같은 선택자에는 마지막에 등장한 선택자가 우선

h1, a {} 로 중복 제거 가능

1. 박스 모델

웹페이지에서 우클릭 > 검사

1. 그리드

어떤 의미도 없고 오직 디자인을 위해 쓰는 태그

<div> </div> - block level element

<span> </span> - inline element

<div id = "grid">

<div>NAVIGATION</div>

<div>ARTICLE</div>

</div>

를 두고 style 태그 안에

#grid{

border: 5px solid pink;

display: grid;

grid-template-columns: 150px 1fr;

}

div{

border: 5px solid gray;

}

fr은 화면 전체를 쓰도록 자동으로 조정되는 단위

1. 반응형 디자인

css를 통해 구현하는 media query

1. 중복되는 css 코드

css.style 파일에 붙여넣고

style 코드 있던 곳에는 <link rel="stylesheet" href="style.css"> 적음

캐싱 – 하나의 웹페이지에서 여러 파일을 바깥에 두고 그것을 다운로드 받는 것과 웹 페이지 안에 css 코드를 내장하는 것 중 네트워크 측면에서는 효율적일까? (적은 트레픽을 사용하는가)

웹 페이지 안에 css 코드를 내장하는 것이 그 자체로는 효율적이라 할 수 있지만 캐싱이라는 테크닉으로 인해 그렇지 않다.

한 번 style.css라는 파일을 다운받으면 파일이 바뀌기 전까지는 웹 브라우저는 이 파일을 우리 컴퓨터에 저장해 두었다가 그 다음에 이 파일을 요청할 때는 저장된 결과를 가져와 속도를 높임(네트워크를 안 쓰기 때문에), 사용료도 적음

**css 문법**

1. 이 웹 페이지에 있는 모든 <a> 태그에 대해서 ~

a {

~

}

2번부터의 css 문법은 전부 ~에 들어가는 내용 – 선언 declaration

a – 선택자 selector

1. 글자 색

빨강으로

color:red;

color – property

red - value

1. 밑줄 제거

text-decoration: none;

style 속성으로 밑줄 추가는 text-decoration: underline

1. 글자 크기

font-size:45px;

1. 가운데 정렬

text-align: center;

1. 테두리 두께

border-width:5px;

1. 테두리 색

border-color:red;

1. 테두리 선 스타일(점선, 실선 등)

border-style:solid;

1. block level element를 inline element만큼 쓰도록

display: unline;

1. inline element를 block level element만큼 쓰도록

display: block;

1. 선택한 태그를 안 보이도록

display: none;

1. Border 중복 제거

border-width:5px;

border-color:red;

border-style:solid;

를 중복을 제거해 아래와 같이 (순서는 상관 X)

border:5px solid red;

1. 콘텐츠와 테두리 사이 간격

padding:20px;

1. 테두리와 테두리 사이의 간격

margin: 0;

1. block level element에서 크기 설정

display: block;

width: 100px;

1. 아래쪽 테두리

border-bottom: 1px solid gray;

1. 특정 id 밑에 있는 ol

#grid ol{

padding-left: 33px;

}

1. 특정 id 밑에 있는 id

#grid #article{

padding-left: 25px;

}

id여서 굳이 이럴 필요는 없지만

1. 반응형 디자인 화면 크기가 800px보다 작다면 사라지도록

@media(max-width:800px){

div{

display:none;

}

}

참고 사이트

1. caniuse.com

현재 웹 브라우저가 해당 기술을 얼마나 채택하고 있는가 확인 가능