Week 12

* css를 적용시키는 방법은 inline, inner ,external이 있다

Inline : 태그 안에 style="내용;"으로 사용하는 방법 style 안에 내용은 css의 문법을 사용해야 한다.

Internal : head태그 안에 <style></style>태그를 넣고 스타일 태그안에 css를 작성하는 방법

External : <link rel='stylesheet' type='text/css' href="css파일 주소" media='all'>를 이용해 외부의 css파일을 불러와 적용시키는 방법(파일 위치는 href에 작성 , 나머지는 몰라도 현재 수업에 상관 없다.)

link방식외에도 import 방식이 있다.

<style type='text/css'>

@import : url('path')

</style >

이 방식으로도 불러 올 수 있다.

이 것의 사용용도는 스타일 태그안에 자신이 적용한  css와 함께 외부 css 파일도 적용 시킬 수 있다.

css문법은 다음과 같다

선택자(selector) { property1 : value;  property2 : value; ...}

선택자는 선택된 태그, 클래스, 아이디 등 해당 선택자에 선택된 것들은 그 다음에 쓴 property(color, backgroud-color ,border...)에 쓴 value값이 적용된다.

inline으로 쓰면 한 컴포넌트? 밖에 css를 적용 못시키지만 inline, external처럼 css를 따로 쓰면 중복도 제거할 수 있고 관리하기도 더 편하다. 밑에서 쓴 것처럼 기본적 위치는 head태그이다.

Selector

Element : 태그 이름 으로 지정 a, p, div, span ,.....

요소명{속성: 값}

class : 클래스 이름으로 지정

.클래스명{속성:값}

id : 아이디 이름으로 지정

#아이디명{속성:값}

선택자의 우선순위는 id > class > element이다.  즉 선택자의 프로퍼티가 적용 될 때 같은 컴포넌트인데 클래스와 아이디의 프로퍼티가 중복되면 아이디의 프로퍼티를 따라가는 것이다.

참고로 id값은 중복되면 안된다. 그냥 해당 컴포넌트를 하나의 객체로 만들어버리는 것이다.  그래서 특히 css를 사용할 땐 컴포넌트들의 아이디와 클래스 값을 지정하는 일이 매우매우 중요하다.

다른 선택자들도 알아보자

전체 선택자

\*{속성 :값} : 전체 요소 선택

가상 선택자

 a : link{속성 :값} - 링크 걸린 텍스트 선택

a : visited {속성 :값} - 링크 방문을 했던 텍스트만 선택

a:hover {속성 :값} - 링크 걸린 텍스트에 마우스를 올렸을 때

a: focus {속성 :값} - 링크 걸린 텍스트에 포커스가 생성되었을 때

a: active{속성 :값} - 링크 걸린 텍스트에 마우스를 눌렀을 때

하위 선택자

ex)

div h2{속성 :값} -div 태그안에 있는 h2태그만 선택

span #id {속성 : 값} - span 태그 안에 있는 id값을 id를 가진 요소만 선택

그룹 선택자

h1,h2,p{속성:값} - h1, h2, p태그 선택

종속 선택자

요소명.클래스명{속성:값} - 선택한 요소 중 지정 클래스를 갖는 요소만 선택

요소명#아이디명{속성:값} - 선택한 요소 중 지정 아이디를 갖는 요소만 선택

css comment

/\*내용 \*/ 형식으로 주석을 단다.

css color

컬러를 들어가기전에 구글 크롬 앱스에 들어가 color picker와 아톰 패키지에서 color picker를 다운 받아서 사용하는 것을 추천한다.

크롬 앱스의 color picker는 웹 상에 있는 색깔을 클릭하면 해당 색깔의 정보를 주어서 개발 할 때 그대로 사용할 수 있다.(클론 코딩할 때 정말 필요)

아톰 패키지에서도 마찬가지로 색을 선택을 하면 자동으로 rgb나 hex코드로 바꾸어줘 개발 할 때 정말 편하고 또한 해당 색깔을 쓰면 그 색깔을 편집기 위에서 표시해준다.

https://khlug.org/image/1985b0ad1eb40713.pnghttps://khlug.org/image/9915b0ad1eb40575.pnghttps://khlug.org/image/1465b0ad1eb40506.png

css 컬러 값(value)은 기본적으로name, rgb, hex코드를 쓴다.

name은 lightblue, red, cyan, yellow, 등 색깔의 영어이름 을 써서 표현하는 것이다.

rgb(red{0~255},green{0~255},blue{0~255},transparent{0~1})가 기본적인 형태이다 transparent는 투명도를 정해준다.

hex  코드는 #XXXXXX 형태로 6자리를 16진수로 표현하는 방식이다.

컬러 표시하는 방법을 알았으니 다음으로 css에서 많이 사용되는 속성 몇가지만 정리하려고한다.

글자체

* + font-family : value; -모양 (글꼴 [구글 폰트 제공 API](https://fonts.google.com/) 에 들어가면 많은 폰트와 적용시키는 방법까지 알려준다. 꼭 들어가보는 것을 추천한다.)

* + font-size : value; - 글자크기 (px, %...단위)

* + font-weight : value; - 글자 두께 (bold , normal ...)

* + font-style : value; - 글자 기울기 (italic , normal...)

* + font : [두께 , 스타일] [사이즈] [글꼴] - 함축 방식 쓴 순서대로 value값 지정한다.

문단 속성

* + color : value; - 텍스트 색상 (name , rgb , hex)

* + text-align : value; - 블록 요소 내에 인라인 요소를 정렬(블록 요소에만 적용)(center, left, right ...)

* + vertical-align : value; - 인라인 요소간의 세로 정렬(인라인 요소에 직접 적용)(top , middle , bottom ...)

* + text-indent :value; - 블록 요소 내에 텍스트 내어/들여 쓰기(px)

* + text-decoration : value; - 텍스트 꾸미기(underline)

배경 속성

* + background-color : value; - 배경 색상(name, rgb, hex)

* + background-image : value; - 배경 이미지 (url('이미지경로'))

박스 모델 (중요)

* + width/height : value; - 가로(너비)/세로(높이) (px)

* + margin/padding : value; - 바깥쪽 여백 / 안쪽 여백 (px)(margin-top, margin-right, padding-bottom, padding-left처럼 상하좌우 한쪽만 선택해서 여백 크기를 조정할 수 있다.)

또한 위치를 지정해주지 않는 경우

2개를 쓰면 margin : apx bpx;( 상하 a픽셀 , 좌우 b픽셀 적용)

3개를 쓰면 margin : apx bpx cpx;(상 a픽셀 좌우 b픽셀 하 c픽셀 적용)

4개를 쓰면 margin : apx bpx cpx dpx;(상 apx 우 bpx 하 cpx 좌 dpx 적용)

padding 도 똑같이 적용

* + border-style : value; - 선 모양(solid, dotted, dashed....)

* + border-width : value; - 선 두께(px)

* + border-color : value; - 선 색깔(name, rgb, hex)

* + border : [두께][스타일][색깔] - border 함축 방식 쓴 수서대로 value적용

display속성

* + display : value; - 요소의 성격을 바꿀 때 사용 ( inline, block, inline-block(인라인 속성에 너비/높이 값 사용 가능), none)

float 속성(블록 요소를 왼쪽, 오른쪽 정렬할 때 사용 ,float를 사용했으면 clear를 통해 float해지하자)

* + float : value; - 블록 요소 좌우 정렬 (left, right)

* + clear : value; - float 해지 할 때 사용(left, right, both)

포지션 - 위치 지정

position : value; - 위치를 지정(relative(현재 위치 기준 위치 지정), fixed(스크린 기준 위치 고정))

요즘 css는 트위터가 jQuery를 이용해 만든 bootstrap을 대부분 사용한다. 위에 내용은 기본적으로 알고 있고 제대로 공부할 거면 bootstrap 공부를 추천한다.

bootstrap에 대해 더 공부하고싶다면 jQuery 공부도 같이하는 것을 추천한다.

* 박정진#39425

2018.05.27. 23:18:43 (99일 전)   
수정 2018.05.28. 22:28:36 (98일 전)

[이동](https://khlug.org/group/35951) [수정](https://khlug.org/group/35951) [삭제](https://khlug.org/group/35951) [보관](https://khlug.org/group/35951)

들어가기 전에

<input type='submit'>과 <button></button>의 기능은 같다. 기본적으로 버튼 태그는 type을 명시하지 않으면 submit 기능을 한다. 즉 둘 다 폼 태그안에서 쓰이며 정보를 전송할 때 쓰이는 태그이다

button 태그의 속성을 알아보면

type='submit' - 폼의 정보 전송 기능

type='reset' - 폼 작성내용 초기화

type='button' - div, span처럼 기능을 아무것도 없이(?) 하지만 자바스크립트 기능을 이용할 때 (event) 자주 사용하는 속성

formtarget - form에서 할 수 있지만 따로 분류할 수 있기 때문에 더 다양하게 폼 정보 전송을 제어할 수 있다.

formtarget="\_self" 현재 창에서 이동

formtarget="\_blank" 새로운 창 생성

기능이 똑같은 태그가 중복되서 있는데 이렇게 구분지은 이유가 있다.

input 에 비해 button 은 css를 적용하는데(꾸미는데) 자유롭다. bootstrap을 사용해 다양한 종류의 버튼을 꾸미고 제작하여 사용할 수 있다.

또한 태그인 만큼 버튼 태그안에 다른 태그도 삽입할 수 있다.

input은 input 태그 답게 입력하는 형식이 자유롭다. 또한 html5로 개발이 되고 다양한 입력 필드를 만들어줘서 개발자가 개발하기 더 편해졌다.

그럼 input 태그에 대해 더 알아보자.

input : form 태그 안에서 사용자의 정보를 입력받는 정말 중요한 태그(파이썬 input매서드와 똑같다고 보면 된다.)

type, value, name 속성이 있는데 세가지 모두 중요하다.

* + type은 어떤 input을 받을 것인지 유형을 결정해주는 속성이다.
  + value는 입력태그의 디폴트값(사용자 변경 가능) , 폼에서 정보를 넘길 때 queryparameter의 값(인자?)
  + name은 폼에서 정보를 넘길 때 서버로 전달되는 queryparameter(변수)의 이름(사용자 변경 가능)
  + form기능을 배우는 week13때 더 자세하게 설명해야겠다.

type은 html5로 넘어가면서 정말 다양한 속성이 생겼다.

* + type='text' :

텍스트 필드로 입력을 받는 유형이다. 특징은 따로 없고(글을 입력하는 창이니까?) value값을 적으면 디폴트 값이 되어 value값에 지정한 내용이 기본으로 써있는 상태에서 출력되게 된다.

참고: (같이 자주 쓰이는 속성 정리)

value : 텍스트 필드에 기본적으로 입력되어있는 내용을 만들 수 있다.

placeholder :  이 속성은 value와 달리 미리 써있는 상태가 아니라 아무것도 안 쓰여 있으면 흐린 글씨로 써있어 클릭하면 내용이 사라지고 빈칸이 되는 기능이다. 사용자에게 입력양식을 알려줄 때 많이 쓰인다.

size : 텍스트 필드의 길이를 지정 (영어가 1글자 한글이 2글자)

maxlength : 텍스트 필드에 입력할 수 있는  최대 문자 개수 를 정해준다.

* + type='password'

기본적으로 text와 같지만 다른 점은 입력값이 ● 이런식으로 보여 사용자도 못보게 된다는 것이다.  단 text처럼 value값을 넣어 기본값을 설정하지는 못한다.

참고 :

size

maxlength

pattern : 허용하는 입력식을 regex(정규표현식)으로 표현해 개발자가 지정한 형식 대로 써야지만 제대로 처리한다.(ex 숫자 + 영어 대문자 + 영어 소문자 순서로 입력해야만 인정(?)을 해주게 만들 수 있다.

* + type='radio'

라디오 버튼이다. 라디오 버튼은 동그라미 체크 형식을 갖고 있고 중복 선택이 불가능 하다.  한 질문에 객관식으로 답을 선택하게 하는 유형일 때 (설문조사)  자주 쓰이는데 중요한 것은 한 질문에 여러가지 라디오 버튼을 만들 때 그 항목들의 name값을 같게 해야 같은 그룹으로 인식을 한다.

checked라는 속성을 이용하여 초기 상태일 때 어떤 값이 미리 선택되있게 할 수 있다.(디폴트 값을 지정할 수 있음!)

* + type='checkbox'

체크 박스다.  라디오 버튼과 똑같고 다른 점은 중복 선택이 가능 하다는 것이다.

* + type='hidden'

사용자에게 보이지 않게 하지만 서버로 전송해야할 경우 사용한다.

* + type='submit'

폼 태그 안에서 작동하는데 폼에서 action으로 지정한 페이지로 폼안에 입력된 정보들을 넘기는 기능을 한다.

* + type='reset'

폼 태그 안에서 작동하는데 폼 안에 입력한 정보를 초기화 시켜준다.

* + type='number', type='range'

number는 스핀박스, range는 슬라이드 막대를 이용하여 숫자를 입력 하게 하는 유형

참고 :

value : 초기값

min : 입력 가능한 최솟값을 지정

max : 입력 가능한 최댓값을 지정

step : 숫자의 간격을 지정 (생략하면 기본 1로 지정이 된다.)

* + type='date', type='month', type='week'

날짜를 입력하는 필드 생성

date - yyyy/mm/dd

month - yyyy/mm

week - 1월 첫째 주 기준 연도 와  주차 수 표시

* + type='datetime', type='datetime-local'

datetime - yyyy/mm/dd 와 시간까지 표시

datetime-local - 자신이 있는 지역의 날짜와 시간 표시

* + type='time'

 시간 지정하는 필드 생성

시간 00:00 ~23:59까지 지정가능 스핀박스로 입력

참고 :

value : 초기값 지정

min : 최솟값 지정

max : 최댓값 지정

step : 스핀박스를 이용해 입력할 때 증가/감소 값 지정

* + type='color'

색 선택하는 필드 생성

참고 :

list : 선택할 수 있는 색상리스트 지정했을 때 그 리스트의 id값 부여

value : 초기값 지정 hex(16진수)코드로 표시

* + type='button'

div, span처럼 아무 기능 없이 일반 버튼 형식 이 된다. 자바스크립트 이벤트와 연동해 많이 사용한다.

* + type='file'

파일을 첨부하려고 할 때 쓰이는 유형

파일선택 ,찾아보기, 선택 등으로 표시

* + type='search'

검색어를 입력하는 필드 생성

* + type='url

url을 입력하는 필드 생성

* + type='email'

이메일 주소를 입력하는 필드 생성

* + type='tel'

전화 번호를 입력하는 필드 생성

* + type='image' src='path' alt=' 대체 내용'

이미지를 전송 버튼으로 만든다.

참고할 만한 다른 속성

readonly : 해당 필드를 읽기 전용으로 바꿈 boolean 형식을 갖고있기때문에 true,false로도 지정 가능

autofocus : 페이지가 로드되면 폼의 요소중 원하는 필드에 마우스 커서가 표시되도록 함

autocomplete : 자동완성 기능 autocomplete='on/off' 로 자동완성 기능을 키고 끌수 있다.

required : 필수 입력 필드 지정하기 readonly와 마찬가지로 boolean형식을 갖고 있기 때문에 true,false로도 지정 가능

부록 – 생활코딩

5일차

css

html언어

<font></font> 글자의 스타일을 바꿀 수있는  태그

 속성 : color="색깔" - 글자의 색깔을 바꿀 수 있다

html은 정보를 전달하기 위한 목적 css는 디자인에 목적을 두어서 서로의 기능을 더 강화 할 수있는 효과를 가져왔다.

<--! 내용 -->  : 주석 기능

<style>내용</style> 스타일 안에 있는 내용은 CSS로 해석해서 처리하게끔 하는 태그

내용 -> 이 안에 CSS언어가 들어간다

ex) a(모든 a태그를) {color(property) : red(value);}(색을 빨강으로)

CSS를 쓰는 방법 2가지

<   style(속성) : CSS 언어로 해석  >

<style> X {} </style> : 선택자 x(selector)에게 {}(declaration)안에 있는 효과를 준다

{속성} : text-decoration : 선택자의 문자 디자인에 대해 설정하는 속성

;을 붙이는 이유는 서로를 구분하기 위해 존재 (효과를 지정한 다음에는 항상 세미콜론 넣기!)

{font - size : @px;} 글자 크기 설정하는 속성

{text-align :  @}  문자의 정렬 방식을 설정하는 속성

<class=" " >" " 라는 클래스를 설정한다.

ex)스타일 태그안에 선택자로  그냥 saw라고 쓰면 saw라는 이름의 태그를 의미

.(클래스 선택자)saw는 클래스가 saw인 것을 선택하는 선택자

<id=" " > 아이디를 설정한다.(클래스와 우선순위를 결정하기 위해서 쓰임)

<style>태그 안에서 id는 #(아이디 선택자)으로 표시한다.

아이디 선택자(중복되지 않아야 한다)>>>> 클래스 선택자 >>> 태그 선택자

만약에 똑같은 선택자이면 가장 나중에 등장하는 선택자의 우선순위가 높다

구체적인 아이디 가 포괄적인 태그 보다 우선순위가 높은것이다(구체적인것이 우선순위가 높다)

6일차

h1 태그는 자동으로 줄 바꿈이 되는데

a태그는 같은 줄에 표시된다.

border-width : -> 박스의 가로 크기

border-color : -> 박스 테두리의  색깔

border-style : -> 박스 테두리의 스타일(실선, 점선 등등)

h1과 a에 테두리를 칠해보면 h1은 화면 전체를 쓰는데 a는 자기 자신의 컨텐츠만 쓰는 것을 알 수있다.

/\*내용\*/ :  CSS 주석

h1 처럼 화면 전체를 쓰는 태그는 block level element(tag)

a처럼 자기 크기만큼만 쓰는 태그 inline element라 한다.

block level element도 자기 크기만큼 쓰는 태그(inline element)처럼 보이게 할 수있다.

display : inline;을 쓰면  inline element가 보여지는 것 처럼 속성을 바꿀 수 있다.

inline element도 마찬가지로 block level element 처럼 보이게 할 수있다.

display : block; 을 쓰면 block level element가 보여지는 것처럼 속성으 바꿀 수 있다.

박스에 관해서는 border라는 속성을 쓰면 된다.

즉, border-width, border-color, border-style처럼 나눠서 쓰지 않고

border : width style color;(속성 순서 상관없이 border에  관한 설정들을 쓰면 된다.)

박스안의 내용과 박스와의 여백을 주고싶으면

padding :  값;

박스끼리의 공간을 띄우고 싶다면

margin : 값;

block element가 화면 전체를 쓰는것을 바꾸고 싶다면

display block;을 쓰고(원래는 display:block;이 생략된 상태)

그 다음에 width : 값 을 쓰면 해당 값 만큼 박스의 크기가 조정된다.

border-위치 : -> 박스의 위 아래 좌, 우 를 쓰면 그 쪽 부분만 조정할 수있다.

웹브라우저에서(크롬) 오른쪽 버튼을 누르면 inspire라고 검색 이라는 항목이 있다. 거기서 개발자 모드로 margin padding border 값 모두 확인 가능하니 보면서 필요한 작업들을 수행하면 될 것같다.

7일차

그리드

<div></div>: 아무런 의미가 없고 디자인으로 이용되는 태그 (block indent level의 태그여서 줄바꿈이 자동으로 된다.)

<span></span> : 아무런 의미가 없고 디자인으로 이용되는 태그(inline level의 태그여서 한 줄에 모두 쓰여진다.)

display : grid; 그리드 속성으로 설정

grid-template-columns : 열을 기준으로 그리드 영역을 나눌 수 있다.

2개면 2개의 크기를 적고 3개면 3개의 크기를 적는다.  크기중  xfr은 전체 화면을 n등분 했을 때   x등분 만큼 사용한다는 것이다. (사용법을 글로 표현을 못하겠다.)

caniuse라는 사이트를 들어가면 이런 언어들을 사용하면 호환이 되는지 를 보여준다. 웹페이지를 제작할 때 참고하면 유용한 사이트인 것 같다.

<https://www.w3schools.com/css/css_grid_item.asp>

<https://developer.mozilla.org/ko/docs/Web/CSS/CSS_Grid_Layout/%EA%B7%B8%EB%A6%AC%EB%93%9C_%EB%A0%88%EC%9D%B4%EC%95%84%EC%9B%83%EC%9D%98_%EA%B8%B0%EB%B3%B8_%EA%B0%9C%EB%85%90>

두 개 사이트를 보면서 그리드에 대해 더 알아보고있다.

<dl></dl> 부모 태그

<dt></dt> 자식 태그 인라인 요소

<dd></dd> 자식 태그 블럭 요소

JavaScript

자바스크립트도 프로그래밍 언어이다보니 반복문, 조건문, 함수 등 파이썬과 동일하지만  문법이 약간 다른 것일 뿐이다. 어차피 웹공부도 하고있는 겸 정리를 해봐야겠다.

내가 볼땐 현재 짧은 기간안에 가르치려다보니 일단 똑같이 쓰고 외워라 라는 식인데, 내 글이 웹 공부하고 있는 사람에게 도움이 되었으면 한다.  똑같이 쓰라고 하는 저 코드에서 왜 저 코드가 들어가는지 의미를 아는게 중요하다고 생각한다. 일단 기말이 있으니까 배우고 있는 거를 순서대로 정리해봐야겠다.

일단 자바스크립트도 html위에서 노는 언어다 보니  css 처럼  inline, inner, external 방법으로 적용 할 수 있다.

1. inline :  inline은 많이 안쓰이는데 많이 쓰이는 부분이 event를 발생시킬 때 onclick, onblur, onmouseover, 등등 이벤트를 다룰 때 자세하게 설명하겠지만 이 이벤트 에 쓰는 문법은 자바스크립트 문법으로 써야한다.

2. inner :  <script type=text/javascript></script> script태그를 써서 그 안에 자바스크립트를 쓰는 것이다. 스크립트 태그 안은 자바스크립트 문법으로 써야하고 자바스크립트 문법으로 인식한다.

3.external : <script type=text/javascript src='~.js'></script> 이렇게 src 속성을 써서 외부의 js 파일을 불러와 쓰는 것이다. css처럼 마찬가지로 따로 외부로 파일을 불러 써야 좋다. 관리의 편리성과 재사용이 용이하기 때문이다. 또한 캐싱면에서도 좋은데 외부로 따로 관리를 하면 html위에 써있을 때와 달리 페이지를 리로드 할때마다 다시 읽어야하는 것이 아니라 한번 읽었으므로 페이지를 리로드할 때 미리 읽었던 파일을 꺼내기 때문에 전송할 때 경량화가 되어 좋다.

그리고 위치에 대해서도 말을 하자면 css는 헤드 태그 안에 js는 바디 태그 끝에 쓰는 것이 네트워크 체감 속도 으로 좋다.

html도 위에서 차례로 코드를 읽어가며 코드를 수행하는데 헤드에 무거운 js파일 이 있으면 페이지의 글자가 로드 되기 전에 js파일을 불러오면 js파일이 크기 때문에 페이지가 뜨는 시간이 길어져 사용자는 로드 되는 시간이 느리다고  인식하겠지만 맨뒤 바디 태그 끝에 쓰면 페이지 가 보여지는 부분을 모두 로드 하고 js파일을 로드하면 사용자는 일단 기능과 시각적인 것들이 눈에 보이므로 페이지 로드 시간이 빠르다고 느끼고 편의성을 느낄 것이다. (css=눈에 보이는 파일, js= 사용자가 기능을 사용할 때 사용하는 파일)

그리고 웹 작업을 할 때 구글 크롬을 사용하는 것이 좋다. 그 이유는 크롬에서 제공하는 inspect(검사) 기능 때문인데 개발자 모드 인지 자세하게는 잘 모르겠다. 여튼 이 항목에 들어가면

css제작, console, network등 개발자가 개발 할 때 필요한 기능을 모두 갖춰져 있다.  그리고 문법 검사와 디버깅도 해주기 때문에 꼭 연습 할 때 자주 이용해야 한다.

처음 나온것이 function이다. function은 파이썬 def와 정말 똑같은 기능을 하고 문법도 똑같다고 생각이 든다.(사용법도 같다, return문을 만나면 함수를 빠져나가는 것도)

function 함수명(parameter){

    block

    return ~

}

다음 으로 나온것이 document.getElementById('id).innerHTML이라는 코드인데 이것을 알려면 DOM(Document Object Model) - 문서 객체 모델을 알아야한다.

파이썬에서 클래스라는 것을 배우면서 클래스명.프로퍼티 혹은 매서드 이런식으로 클래스의 프로퍼티, 매서드를 불러서 사용한 것을 기억해보면 document라는 클래스안에 getElementById라는 프로퍼티 혹은 매서드를 불러왔다고 생각하면 쉽다.

일단 기본적으로 파이썬에서도 object가 최상위 객체 였고 클래스면 모두 object를 상속 받은 클래스 이므로 쓸 때 생략해도 되는 것을 알고있다.

자바스크립트도 마찬가지다.

document객체는 window 객체의 속성이므로 사용할 때 window.document라고 사용해야하지만 window는 생략할 수 있다.

dom에 대해 자세하게 다루려면 끝이 없을 것 같다, 내가 느끼기에 일단 필요한 것만 정리를 해야겠다.

선택자에 대해서 일단 말하면 직접 선택자(절대경로라고 생각) 인접 선택자(상대 경로라고 생각) 이있다.

직접 선택자

* document.getElementById('id명') - 해당 아이디를 가진 요소를 선택
* document.getElementByClassName('class명') - 해당 클래스 명을 가진 첫번째 요소 선택
* document.getElementByTagName('태그명') - 해당 태그명을 가진 첫번째 요소 선택

두번 째 세번 째 요소를 선택하고 싶을 때는 해당 매서드 뒤에 리스트 처럼 [0],[1],[2] ... 를 쓰면 된다. 첫번 째 요소는 0인데 생략 가능하고 두번째 요소를 선택하고 싶으면 [1]을 붙여주면 된다.

ex) getElementByTagName('p')[1] .~

인접선택자는 수업 시간에 다루지 않으니 나도 다루지 않겠다. 알려면 문서 객체 구조를 알아야되고 설명할게 너무 많아질 것 같다.

CSS 선택자 를 이용하여 선택할 수 있는 방법도 있다.

* document.querySelector('#id, .class ,...') 해당 아이디, 클래스명 등을 가진 첫번 째 요소 선택
* document.querySelectorAll('#id, .class ,...') 해당 아이디, 클래스 명 등을 가진 모든 요소 선택

속성 관련 매서드

document. 선택자. 속성명 - 선택한 요소의 지정한 속성값을 리턴

document.선택자.getAttribute('속성'); - 선택한 요소의 지정한 속성값을 리턴

document.선택자.속성명='value' - 선택한 요소의 지정한 속성값을  value값으로 수정 혹은 생성

document.선택자.setAttribute('속성','value'); - 선택한 요소의 지정한 속성값을  value값으로 수정 혹은 생성

innerHTML 속성  매서드

document.선택자.innerHTML;  - 선택한 요소의 모든 하위 요소를 문자 데이터로 리턴

document.innerHTML='value'; - 선택한 요소의 모든 하위 요소를 value값으로 수정 혹은 생성

document.write('내용'); - 내용을 출력한다

document.writeln(); - 내용 한줄을  출력한다

주의할 점 : document.write를 페이지가 로드가 끝난 후 핸들러나 콘솔을 통해서 실행 시키면 이전 페이지 내용은 모두 지워지고 document.write로 쓴 내용으로 대체된다.

document 객체 프로퍼티

document.title -  title 요소값을 반환

document.body - body 요소값을 반환

document.head - head  요소 값을 반환

document.forms - form 요소 목록을 반환

document.images - 이미지 요소 목록 반환

document.anchors - a 요소 목록 반환

document.links - 링크 요소 반환

document.scripts- script 요소 값 반환

이외에도 window , screen, location 등 다양한 객체들이 있다.

이벤트 - 유저가 페이지 에 행하는 모든 동작

이벤트 종류

마우스

onmouseover - 마우스(커서)가 지정한 요소에 올라갔을 때 실행

onmouseout - 마우스가 지정한 요소에서 벗어났을 때 실행

onclick - 마우스가 지정한 요소를 클릭했을 때 실행

ondbclick  - 마우스가 지정한 요소를 더블클릭했을 때 실행

키보드

onkeypress - 지정한 요소에서 키보드를 눌렀을 때 실행(shift, ctrl, 등 기능 키 및 방향키 제외)

onkeydown - 지정한 요소에서 모든 키가 눌렸을 때 실행

onkeyup - 지정한 요소에서 키보들 눌렀다 떼었을 때 실행

기타

onfocus - 지정한 요소에 포커스가 갔을 때 실행

onblur - 지정한 요소에서 포커스가 생겼다 잃었을 때 실행

onload - 지정한 요소의 하위 요소를 모두 로딩 했을 때 실행

onunload - 문서를 닫거나 다른 페이지로 이동할 때 실행

onsubmit - 폼 요소에서 submit 버튼을 눌렀을 때 실행

onreset - 폼 요소에서 reset버튼을 눌렀을 때 실행

이벤트 핸들러 - 이벤트 발생시 자바스크립트 실행문을 실행하는 것

이벤트 등록 방식은 두가지이다

직접 요소 이벤트 등록 : 태그에 직접 등록 하는 방법

ex <input type='text' id='myevent' onblur='functionName'>

 DOM을 이용한 이벤트 등록  : script 나 js 파일에 등록하는 방법

ex)  <input type='text' id='myevent'> html에 작성

script 내부나 js 파일에

document.getElementById('myevent').onblur=function(){ code;}

window 객체는 브라우저의 객체 (BOM의 최상위 객체 이다.)

window 객체는 document, screen, location, history, navigator 하위 객체를 포함하고 있다. 이들을 통틀어 BOM이라고 한다.

window 객체의 메서드 (window는 브라우저에 내장된 객체의 최상위 객체이므로 따로 앞에 window.alert() 처럼 window를 붙이지 않고 alert()만 작성해도 된다.)

open('url 경로','창 이름', '옵션') - 새 창을 열 때 사용

alert('내용') - 경고 창을 띄움

prompt('내용', 기본default답변) - 질의응답 창을 띄움

comfirm('내용') - 확인/취소 창을 띄움 (확인이면 true 취소 면 false 값 리턴)

자바스크립트의 css 조작

document.선택자.style.~

[자바스크립트 css 스타일 조작](https://www.w3schools.com/jsref/dom_obj_style.asp)

부록 – 생활코딩

8일차

반응형 디자인과 미디어 쿼리

@media{조건(?){CSS 조건}

ex){조건} :  min-width:800px(세미콜론 안 붙인다) -> 화면의 크기가 800px보다 클 때

       max-widht:800px -> 화면의  크기가 800px보다 작을 때

html css를 먼저 불러오고

javascript는 되도록 바디 끝(파일을 불러올 때 화면을 보여주고 불러오는게 더 좋음)

파일을 따로 쓰는게 캐쉬를 통해서 속도의 향상, 전송량의 경량화에 도움이 됨.

<link> : css파일을 불러와 해당 파일에 css파일이 적용 되게 하는 태그

JavaScript

javascript 를 쓰는 방법 3가지

1.<script></script> 태그 이용하기 : <script>태그 안에 있는 내용은 js문법에 맞게 컴퓨터가 해석하게 하는 태그이다.

2.event이용하기 :

<input>태그 : 사용자의 입력을 받는 태그

속성 : type="button" -> 버튼입력형식  value ="내용"(버튼에 내용을 써준다.)   type="text" -> 텍스트 입력형식

on~ : on뒤에 나오는 내용은 javascript가 나와야한다.

           on~을 실행했을 때 on~뒤에 있던 내용을 기억했다가 js로 해석해 실행한다.

onclick -> 버튼을 클릭했을 시

onchange -> 내용이 변경됬을 시

onkeydown -> 키를 눌렀을 시

alert:  알람?을 띄운다.

3.console창 이용하기

오른쪽 버튼에 inspect라는 항목을 누르면 element옆에 콘솔창이 뜨는데 콘솔창으로 html위에 javascript로 제어 및 실험(?)을 할 수있다.

물론 콘솔창에 쓴 것은 저장되지 않는다.(터미널이나 파이썬 shell창과 비슷한 개념인 것 같다.)

str.length -> 문자열의 개수를 출력한다.

[데이터타입 - 문자열과 숫자](https://opentutorials.org/course/3085/18870)

[변수와 대입 연산자](https://opentutorials.org/course/3085/18871)

자바스크립트 데이터 타입

boolean, Null, Undefined, Number, String, Symbol, object

보면 파이썬과 문법이랑 기존 클래스가 갖고있는 매서드 형식도 비슷하다. 그래서 빠르게 넘길 수 있는 것 같다.

차이점이 있다면 끝에 ;을 붙이는 것이 다른 점인 것 같다.

관련 자바스크립트 문법을 다 정리해놓은 moz://a 라는 사이트를 자주 이용해야겠다

[developer.mozilla.org/ko/docs/Web/JavaScript](https://developer.mozilla.org/ko/docs/Web/JavaScript)

CSS기초

CSS 수업에 있는 내용이라 빠르게 복습하고 넘어갔다

제어할 태그 선택하기

document.querySelector(선택자(ex .myclass, #myid, "태그)")

document.getElementById((선택자(ex .myclass, #myid, "태그"))

둘다 자바스크립트에서 제어할 선택자를 지정하는 함수이다.

document.querySelector(선택자(ex ".myclass", "#myid", "태그)").style.속성;

style뒤에 들어가는 문법도 자바스크립트의 문법이니 CSS와 헷갈리지말자

프로그램, 프로그래밍, 프로그래머

프로그램 ->  순서  
순서를 만드는 과정 -> 프로그래밍  
순서를 만드는 사람 -> 프로그래머

조건문

토글(toggle)을 만든다

비교연산자 과 boolean

파이썬과 비교연산자는 똑같다. 다른점은

===인데 이것은 값이 같고 데이터 형태도 같아야 True를 리턴한다.

!==도 있는데 값이 다르거나 데이터 형태가 다르면 True를 리턴한다.

[자바스크립트 비교연산자 정리](https://developer.mozilla.org/ko/docs/Web/JavaScript/Guide/Expressions_and_Operators)

논리연산자 !(not) &&(and) ||(or)

tip : html에서는 <, >가 하나의 문법이므로 이걸 표시하려면 &lt(<), &gt(>)를 쓰면된다

조건문

document.write("문자") (파이썬에서 print)

자바스크립트에서 boolean은 파이썬처럼 맨 앞 글자가 대문자가 아니라 true, false로 구분된다.

if(boolean){수행할 문장} -> true일 때 실행된다.

9일차

자바스크립트가 생각보다 많이 어렵다 일단 생활코딩 강좌는 한번 쭉 보고 실습한 후 책에 있는 예제로 더 공부해봐야겠다.

node.js 나가기에는 내가 아직 자바스크립트에 대한 이해도가 조금 떨어진다는 생각이 든다.

시험이 끝나면 올렸던 것들도 한번 정리가 필요한 것 같다.

js는 문장 끝에 ;을 붙여줘야한다.

조건문의 활용 if(조건){ }

리팩토링

this : 해당 블록의 자신을 가르키는 객체 (조금 더 파헤쳐 봐야겠다)

반복문 예고

배열

자바스크립트에선 변수를 선언할 때 var을 붙인다.

파이썬에서 리스트와 같은 개념인 듯 하다. (인덱스도 처음에 0부터 시작)

.push() : 리스트에 요소를 추가할 수 있다.

.length : 리스트 요소, 문자 의 개수를 리턴한다

반복문

for문과  for..in 문 , while문 이 있다.

while(조건이 참일 동안){수행할 문장;} : 조건이 거짓이 되면 루프를 빠져 나온다.

일단 이 또한 파이썬과 비슷하다.

함수

function 함수이름 (매개변수) {

return  ;

}

파이썬의 def와 같음

10일차

객체  (파이썬 클래스)

객체안에 있는 변수? 프로퍼티에 대해서는 조금 더 알아봐야겠다(강의에서는 딕셔너리 처럼 key값에 value값을 대응하는 형식으로 했다.)

객체 안에서 작동하는 함수 -> 매서드( 파이썬과 똑같음)

<script src="주소"></script>

js도 마찬가지로 html과 분리해서 따로 관리하는 편이 여러모로 괜찮다.

절대주소  : 파일이나 url([http://](NULL) 부터 시작)의 전체적 주소

상대주소 : 상위 디렉터리  -> ../

       하위 디렉터리 -> /

       현재 디렉터리 -> .(생략 가능)

상대주소를 쓰는 편이 이식성 부분과 수정할 때 매우 좋은 것 같다.

라이브러리 : 완성된 부품들 -> 부품들을 끼워 맞춰 완성

프레임워크 : 구조 -> 만들어진 구조 속에서 작업을 한다.

UI  : User Interface ->  사용자가 작동하는 기능 -> jQuery

API :  Application Programming Interface ->  프로그램어를 입력하면 동작하는 기능(?)

11일차

<form></form> 입력한 정보를 action이라는 속성을 통해 전송하는 태그  urlquerystring을 생성하는 공장과도 같다.

정보를 가져올 때는 url query string을 이용해 가져오는 것이 맞다.

하지만 생성,삭제를 할 때는 url query string 을 이용하면 나중에 누군가 이런 주소(url query string 을 쓰는)를 쓰면 정보를 마음대로 생성 삭제 가 가능할 수있다.

그래서 form 의 method 속성을 쓰는 것이다.

method : 폼을 서버에 전송할 http 메소드를 정하는 속성. (GET (읽을 때 사용)또는 POST(생성, 수정, 삭제 할 때 사용))

action : 폼을 전송할 서버 쪽 스크립트 파일을 지정하는 속성.

name :폼을 식별하기 위한 이름을 부여