VScode 유틸리티

Live server : 새로고침 없이 결과를 바로바로 보여주는 유틸리티.

**1.HTML**

Html:5 코드로 시작

**시멘틱 태그**

영역을 나누는 태그. <body> 안에 작성한다.

<header>, <nav>, <figure>, <main>, <aside>, <section>, **<article>,** <footer> 8가지가 있다.

<article> : 기사가 있는 공간이다. <div>를 여러 개 묶는다.

**서식 태그**

콘텐츠를 채울 때 사용한다.

<h1~h6>, **<p>**, **<div>**, <strong>, <a>, **<span>**, <img>, <br> 등등이 있다.

<p> : 글을 그룹화한다. 문단이다.

<div> : 콘텐츠를 그룹화 한다. <p> 여러 개를 묶는다.

<span> : 인라인 요소를 그룹화 한다. 즉. 꾸미는 태그만 묶을수 있다. <strong>, <em> 등 만을 묶을 수 있다는 얘기이다.

묶는 단위 : <span> -> <p> -> <div> -> <article>

**목록 태그**

목록을 만들 때 사용한다.

<ul> - <li> : unoriented list. 순서가 없는 목록을 만든다.

<ol> - <li> : oriented list. 순서가 있는 목록을 만든다.

<dl> - <dt><dd> : defined list. 정의형 목록을 만든다.

Dt는 defined title. 제목.

dd는 defined data 내용..

**폼 태그**

사용자 정보를 입력 받는다.

<input type = “text”> : 무언가를 입력 받는다.

속성 값은 “text”, “password”, “checkbox”, “radio”, “file”, “color”, date”, “submit”, “reset” 9가지가 있다.

<select> - <option> : 선택 목록을 만든다.

<textarea> : 텍스트 입력 공간을 만든다.

<form action=”전송 위치”, method=”전송 방법”> : 데이터베이스로 어떤 값을 넘긴다.

묶는 관계 : <input> -> <form>

**멀티미디어 태그**

<audio>, <video>가 있다.

**Quick Result**

Html:5

글 묶는 단위 : <span> -> <p> -> <div> -> <article>

컨텐츠 묶는 단위 : <> -> <section>

데이터 입력 및 전송 : <input> -> <form>

**2. CSS**

**Before Start**

Html // link:css 입력하여 파일 연결

Css // @charset “utf-8”; 입력하여 한글 지원

**Selector**

\* //전체 선택.

P // 태그 선택.

P strong // 자손 선택. 자신과 하위 요소를 전부 선택한다.

P > strong // 자식 선택. 자신의 바로 및 요소만 선택한다.

H1,h2,h3 // 다중 선택.

* Ul li : nth-of-type(1)
* Ul li : nth-of-type(2)
* Ul li : nth-of-type(3) // 순서 선택자. html 태그의 순서를 골라서 선택한다.
* Ul li : nth-of-type(2n) // 수열 선택자. 2,4,6,8,… 순서의 태그만 선택해줄 수도 있다.
* Ul li : last-child // 마지막 요소 선택자.

**input[type=”text”] // 속성 선택자.**

H1:hover //가상 선택자. 마우스를 올렸을 때 효과가 나타나게 해준다.

H1:before

H1:after //가상 선택자. 지정한 태그의 앞 뒤에 특정 컨텐츠를 넣을 수 있다.

**class 선택자(매우 중요) : CSS 스타일링을 위해, 임의의 이름을 붙인다.**

HTML

<p class=”클래스명”>

CSS

**태그명.클래스명** {

CSS 속성

}

**id 선택자(매우 중요) : CSS 스타일링, JavaScript 기능 추가를 위해 이름을 붙인다. 주로 “시멘틱 태그”에 사용한다.**

HTML

<div **id=”아이디명”**>

CSS

**#아이디명** {  
 CSS 내용

}

**CSS의 기능**

글자 관련

Font-size

font-family

color

line-height

text-align

letter-spacing

영역 관련

Width, height

Margin //auto는 “좌우 정렬”을 의미한다.

Padding

Box-sizing //기존 박스 크기를 유지하면서, 안쪽 여백을 지정한다.

Border-width, border-style, border-color, border

인라인 요소 설정

인라인 요소(strong, em, a, span)는 본래 높이와 너비를 설정할 수 없다. 그러나 강제로라도 설정하고 싶다면, display:inline-block; 속성을 추가한다. 이 속성은 인라인 요소와 블록 요소의 속성을 모두 사용한다는 뜻이다.

배경

Background-color

background-image

background-repeat

background-position

background-size

background-attachment

레이아웃

가장 먼저, CSS를 초기화 해준다.  
\* {

Margin : 0px;

Padding : 0px;

}

Ul, ol {

List-style : none;

}

A {

Text-decoration : none;

}

블록 요소 좌우로 배치하기

Float 속성을 사용해준다.

HTML

<div class=”wrap”>

<section class=”left”>

<section class=”right”>

…

CSS

.wrap::after { //부모 요소가 float를 인식할 수 있도록 해준다.  
 content: ‘’;

Display: block;

Cleaer: both;

}

.wrap .left {

Float: left;

}

.wrap .right {

Float: right;

}

블록 요소 임의로 배치하기

Relative, absolute, fixed 속성을 사용해준다.

콘텐츠를 특정 영역에 채우기

Object-fit 속성을 사용한다.

위/아래로 보내기

z-index 속성을 사용한다. 주의할 점은, 동일한 자식 레벨에서만 통한다는 점이다.

그래픽 효과

Opacity

Box-shadow

Text-shadow

Border-radius

Gradient

Filter

콘텐츠 모양 변형

Transform(2D)

Transform(3D)

Perspective

Transform-style

Transform-origin

사용자 행동에 반응하면 전환 효과

Transition-property

Transition-duration

Transition-delay

Transition-timing-function

Transition

자동 에니메이션 효과

@keyframes //에니메이션을 등록한다.

Animation-name

Animation-duration

Animation-timing-function

Animation-iteration-count

Animation-delay

Animation-play-state

Animation

SVG // Scalable vector graphics. 웹에서 벡터 이미지를 업로드 하는게 아닌, 좌표값을 입력 받아서 웹 상에서 벡터 이미지를 그리는 기술을 의미한다.

화면 너비에 따라 웹 페이지 디자인 변경하기

@media screen and (max-width: 1000px) {

CSS 코드

}

**+) FLEX 레이아웃**

Display 속성으로 부모 태그에 flex 레이아웃을 적용할 수 있다.

Justify-content로 자식 태그에 flex 레이아웃 축을 기준으로 컨텐츠를 정렬한다.

Align-content 혹은 align-items 로 자식 태그에 flex 레이아웃 반대 축을 기준으로 컨텐츠를 정렬한다. 전자는 자식 요소가 여러 개고 줄바꿈이 필요할 때. 후자는 자식 요소가 1개거나 줄바꿈이 필요 없을 때 사용한다.

Flex 레이아웃 자식 태그 속성

Order //태그 순서 변경

Flex-grow, flex-shrink, flex // 비율 조정

**3. JavaScript**

**시작하기**

HTML

<body> -> <script> 안에 JavaScript 코드 작성 혹은

<body> -> <script src=”파일 경로”> 혹은

<head> -> <script defer src=”파일 경로”>

**HTML 요소 선택하기(DOM 모델 생성)**

HTML

<section id=”wrap”>

<article class=”box1”>

<article class=”box2”>

<article class=”box3”>

…

JavaScript

Const frame = Document.querySelector(“#wrap”) //ID 선택자 선택

Const box1 = Document.querySelector(“#wrap .box1”) //ID -> Class 선택자 선택

Const items = Document.querySelectorAll(“#wrap article”)

For(let item of items) {  
 console.log(item);

} //article 모든 태그를 선택하고. 이를 반복문으로 하나하나 접근하는 방식

For(let i=0; i<items.length; i++) {  
 console.log(item);

} //이렇게도 사용 가능하다.

**HTML 요소 선택하기(고급)**

HTML

<ul class=”list”>

<li class=”item1”>

<li class=”item2”>

<li class=”item3”>

<li class=”item4”>

…

JavaScript

자식 요소 선택하기

Const list = document.querySelector(“.list”);

Const items = list.children; //배열 형태로 선택자를 저장한다.

Console.log(items); //배열 선택자 출력

Console.log(items[0]); //배열 0번째 선택자 출력

Console.log(items[1]); //배열 1번째 선택자 출력

Console.log(items[2]); //배열 2번째 선택자 출력

Console.log(items[3]); //배열 3번째 선택자 출력

부모 요소 선택하기

Const item2 = document.querySelector(“.item2”);

Console.log(item2.parentElement); //바로 위 부모 선택자 출력

형제 요소 선택하기

Const item3 = document.querySelector(“.item3”);

Console.log(item3.previousElementSibling); //item2 선택자가 출력된다.

Console.log(item3.nextElementSibling); //item4 선택자가 출력된다.

HTML

<body>

<main>

<section>

<article>

<ul>

<li>list</li>

…

JavaScript

최상위 부모 요소 선택하기

Const li = document.querySelector(“li”);

Console.log(li.closest(“main”)) //자신의 부모 중, main이란 선택자를 반환한다.

**CSS 제어하기**

HTML

<body>

<article id=”box”>

…

CSS

#box {

Width: 200px;

Height: 200px;

Background-color: aqua;

Border : 5px solid #000;

Transform: rotate(0deg);

}

JavaScript

Const box = document.querySelector(“#box”)

Box.style.width = “10%”;

Box.style.height = “300px”;

Box.style.backgroundColor = “hotpink”;

Box.style.border = “none”;

Box.style.transform = “rotate(10deg)”;

**Event 연결하기(매우 중요)**

HTML

<body>

<a href=<https://www.naver.com>>

…

CSS

A {

Font-size: 100px;

Color: #555;

}

JavaScript

Const link = document.querySelector(“a”);

**Link.addEventListener(“click”, ()=>{**

**Console.log(“링크를 클릭했습니다.”);**

**} //요소명.addEventListener(“이벤트명”, (전달될 값) => {실행할 구문});**

Link.addEventListener(“click”, (e)=>{

e.preventDefault();

console.log(“링크를 클릭했습니다.”);

}

HTML

<body>

<div id=”box”>

…

CSS

#box {  
 width: 200px;

Height: 200px;

Background: aqua;

Margin: 100px auto;

}

JavaScript

Const box = document.querySelector(“#box”);

Box.addEventListener(“mouseenter”, ()=>{

Box.style.backgroundColor = “hotpink”;

});

Box.addEventListener(“mouseleave”, ()=>{

Box.style.backgroundColor = “aqua”;

});

**+) ES5 버전의 Event 연결하기**

Const box = document.querySelector(#box”);

Box.addEventListener(“mouseenter”, function() {  
 this.style.backgroundColor = “hotpink”;

});

Box.addEventListener(“mouseleave”, function() {  
 this.style.backgroundColor = “aqua”;

});

**이벤트 한번에 연결하기**

HTML

<body>

<ul class=”list”>

<li><a href=”#”>item1

<li><a href=”#”>item2

<li><a href=”#”>item3

<li><a href=”#”>item4

…

JavaScript

Const list = document.querySelectorAll(“.list li”) //배열 형식으로 받음

For(let el of list) {

El.addEventListener(“click”, e=> {

e.preventDefault();

console.log(e.currentTarget.innerText);

클릭 이벤트가 발생할 때 숫자를 증가, 감소하기

HTML

<body>

<a href=”#” class=”btnPlus”>plus

<a href=”#” class=”btnMinue”>minus

</body>

JavaScript

Const btnPlus = document.querySelector(“.btnPlus”);

Const btnMinue = document.querySelector(“.btnMinus”);

Let num = 0;

//btnPlus를 클릭할 떄 마다

btnPlus.addEventListener(“click”, e=> {  
 e.preventDefault();

num++;

console.log(num);

}

//btnMinus를 클릭할 때 마다

btnMinus.addEventListener(“click”, e=> {

e.preventDefault();

num--;

console.log(num);

}

**JavaScript에서 “ “ 안에서 변수 사용하기**

const myName = “홍길동”;

console.log(‘내 이름은 ${myName} 입니다.’);

**JavaScript로 클래스 제어하기**

가급적이면 JavaScript에서 클래스를 제어하는게 낮다. 그 이유는 CSS를 DOM으로 제어하면 CSS를 바꾸는게 아닌, HTML에 인라인 요소로 직접 적용하기 떄문에, CSS 우선순위가 낮아지기 때문이다.

따라서 javascript로 클래스를 제어할 때, 클래스를 추가하여 CSS를 변경하는 작업을 진행한다.

HTML

<body>

<section id=”wrap”>

<article></article>

</section>

</body>

CSS

#wrap.on article { //wrap 요소에 클래스 on이 추가된다.

Background: hotpink;

}

Javascript

Wrap.addEventlistener(“click”, ()=> {

wrap.classList.toggle(“on”);

});

**Javascript 함수**

Function plus(num1, num2) {

Console.log(num1+num2);

);

Plus(2,3);

**JavaScript로 HTML 요소의 속성값 제어하기**

HTML

<body>

<a href=”https://www.naver.com>네이버</a>

</body>

JavaScript

Const link = document.querySelector(“a”);

Const link\_href = link.getAttribute(“href”); //속성값 가져오기

Const new\_href = <https://www.nate.com>;

Link.setAttribute(“href”, new\_href); //속성값 새로 지정

Console.log(link\_href);

**JavaScript로 사용자 브라우저 판단하기**

Const ver = navigator.userAgent;

Console.log(ver); //이러면 익스플로러 브라우저에서는 trident라는 정보가 있다.

익스플로러만 정보를 구별하려면?

Const isIE = /trident/i.test(ver);

Console.log(isIE);

If(isIE) {

Alert(“익스플로러로 접속하셧네요.”);

}

**JavaScript Quick Search**

1. HTML 객체(DOM 요소) 불러오기

Const title = document.querySelector(“#title”);

1. 이벤트 연결하기

Title.addEventListener(“click”, ()=> {  
 console.log(“링크를 클릭했습니다.”);

}

1. CSS 제어하기(CSS 클래스 추가 및 속성 변경)

**HTML**

<body<

<section id=”wrap”> …

**CSS**

#wrap.on article {

…

**JavaScript**

Wrap.addEventListener(“click”, () => {

Wrap.classList.toggle(“on”);

}

1. HTML 속성 가져오기 및 변경하기

Const link = document.querySelector(“a”);

Const link\_href = link.getAttribute(“href”);

Const new\_href = <https://www.nate.com>;

Link.setAttribute(“href”, new\_href);