

# 모바일 웹 React.js

컴퓨터공학과 이병문

2022.9.5-8

01 강의/일정소개, 평가소개

02 리액트 준비/소개

03 리액트시작하기, **JSX**

04 렌더링, 컴포넌트/**Props**

05 **State** 와 생명주기, 훅

06 중간고사

## React 준비(Web기본)

- HTML Review
- CSS Review
- JS Review

## React 소개

- 리액트 소개
- 개발환경 설정

## ■ HTML Review

기본구조

기본태크

입력폼태그

DOM

시멘틱태그

### ■ 웹사이트의 구조

#### ☑ 구조와 요소

- HTML ... 웹페이지내용
- CSS .... 웹페이지형식
- JS ..... 기능과 동작

```
<!doctype HTML>
<HTML>
  <HEAD>
    <style> ..... </style>
    <script> ..... </script>
  </HEAD>
  <BODY>
    <h4> ... </h4>
    <div> .... </div>
  </BODY>
</HTML>
```

CSS →

JS →

HTML → {

### ■ HTML 구조

#### ☑ 문서구조

```
- 헤더/바디: <html>
               <head> ... </head>
               <body> ... </body>
             </html>
```

## ■ HTML Review

기본구조

기본태크

입력폼태그

DOM

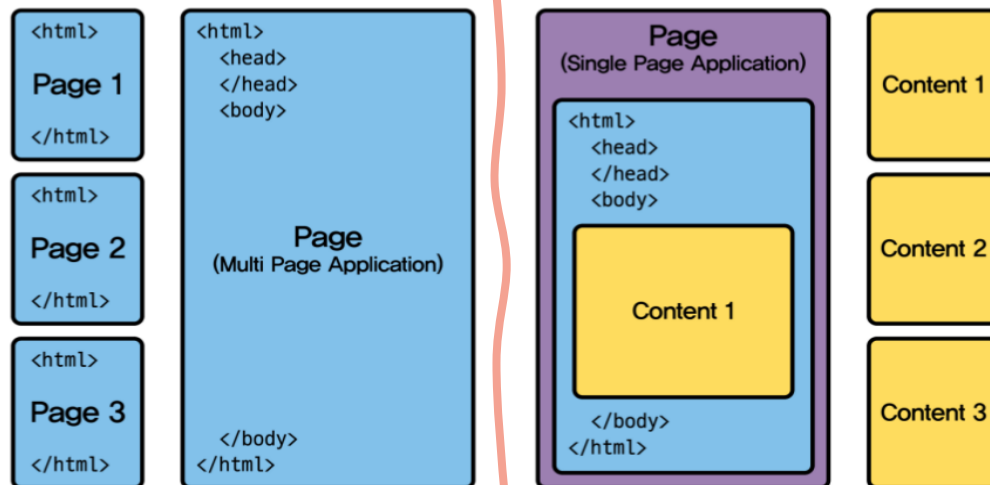
시멘틱태그

### ■ HTML 문서의 구조

#### ☑ SPA

(Single Page Application)

- 하나의 페이지구조에 다양한 콘텐츠를 담아서 구성
- MPA방식의 한계를 탈피



▶ MPA vs. SPA

#### ☑ HTML ... 태그(=엘리먼트)로 표현

태그이름 (엘리먼트)      속성      CSS3 스타일      이벤트 리스너

```
<p id="firstP" style="color:blue" onclick="this.style.color='teal'">  
이것은<span style="color:red">문장입니다.</span>  
</p>
```

콘텐츠(innerHTML)

## ■ HTML Review

기본구조

기본태크

입력폼태그

DOM

시멘틱태그

### ■ 문단관련 태그

- ☑ 제목크기: `<h1 title="풍선 도움말"> 제목 </h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>`
- ☑ 단락나누기 / 블록나누기: `<p> ... </p>` `<div> ... </div>`
- ☑ 기타 : `<br>` `<hr>`

### ■ 문자관련 태그

- ☑ 텍스트 꾸미기: 권고: css 의 property 로 대체
  - `<b>` `<strong>` `<em>` `<i>`
  - `<small>` `<sup>` `<sub>` `<mark>`

### ■ 테이블(표)관련 태그

- ☑ `<table> ... </table>` 권고: css 의 flex , position 등으로 대체

제목	<code>&lt;caption&gt;</code>	1학기 성적		
헤드	<code>&lt;thead&gt;</code>	이름	HTML	CSS
바디	<code>&lt;tbody&gt;</code>	황기태	80	70
		이재문	95	99
		이병은	40	61
바닥	<code>&lt;tfoot&gt;</code>	합	225	230

## ■ HTML Review

기본구조

기본태크

입력폼태그

DOM

시멘틱태그

### ■ 미디어관련 태그

- ☑ 비디오 삽입: `<video src="movie.mp4" controls autoplay> ... </video>`
- ☑ 오디오 삽입: `<audio src="music.mp3" controls autoplay loop> ... </audio>`
- ☑ 이미지 삽입: ``

### ■ 하이퍼링크관련 태그

- ☑ 링크 삽입: `<a href="http://www.gachon.ac.kr"> 가천대 </a>`  
`<a href="myfile.pdf" download> 파일다운로드 </a>`

### ■ 리스트관련 태그

- ☑ 순서표기: `<ol>` `<li>` 첫번째 `<li>` 두번째 `<li>` 세번째 `</ol>`
- ☑ 비순서 표기: `<ul>` `<li>` 첫번째 `<li>` 두번째 `<li>` 세번째 `</ul>`

⋮

## ■ HTML Review

기본구조

기본태크

입력폼태그

DOM

시멘틱태그

### ■ 폼태그

<h3> 회원가입 </h3>

<form>

<table>

<tr> <td> -아이디 </td> <td> <input type="text"> </td> </tr>

<tr> <td> -비밀번호 </td> <td> <input type="password"> </td> </tr>

<tr> <td> -한번더 </td> <td> <input type="password"> </td> </tr>

<tr> <td> -성별 </td> <td> <input type="radio" name="gender"> 여성  
<input type="radio" name="gender"> 남성 </td>

<tr> <td> -관심분야 </td> <td> <input type="checkbox" name="mine1"> 백엔드  
<input type="checkbox" name="mine1"> 프론트 </td>

<tr> <td> -전화번호 </td> <td> <select> <option> 02 </option> <option> 031 </option>  
<option> 032 </option>  
</select> </td> </tr>

<tr> <td> -가입일 </td> <td> <input type="date" value="2022-09-05"> </td> </tr>

<tr> <td> <br> <button type="submit"> 가입 </button> </td> </tr>

</table>

</form>

회원가입

-아이디

-비밀번호

-한번더

-성별 ☐ 여성 ☐ 남성

-관심분야 ☐ 백엔드 ☐ 프론트엔드

-전화번호 02

-가입일 02-09-05 가입일

가입

2022년 09월

일	월	화	수	목	금	토
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

지우기 오늘



## ■ HTML Review

기본구조

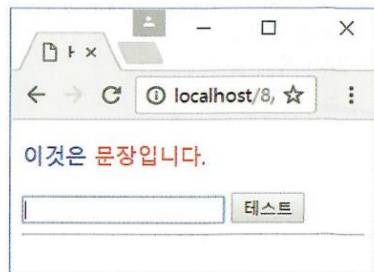
기본태크

입력폼태그

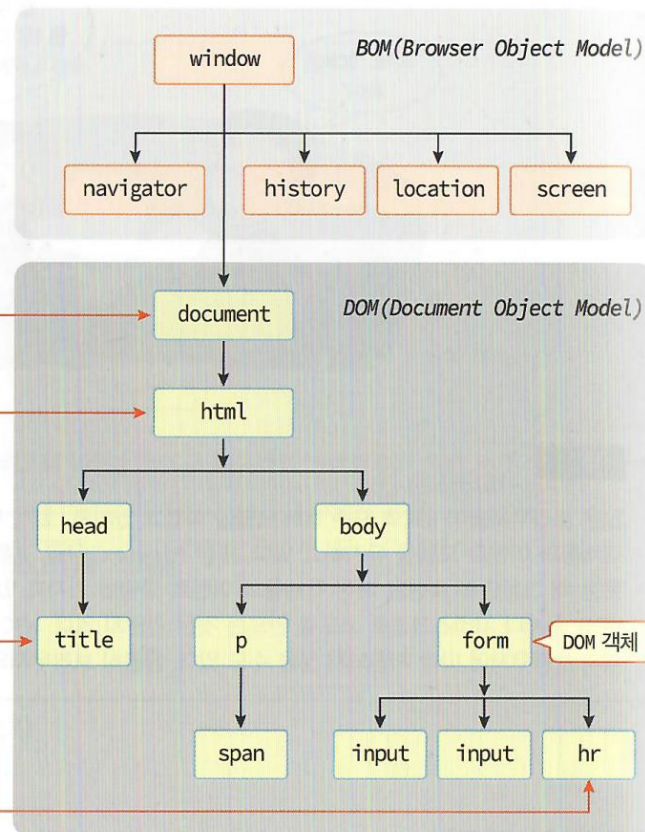
DOM

시멘틱태그

## ■ HTML 과 DOM(Document Object Model)



```
<!DOCTYPE html>
<html>
<head>
  <title>HTML DOM 트리</title>
</head>
<body>
<p style="color:blue">이것은
  <span style="color:red">문장입니다.
  </span>
</p>
<form>
  <input type="text">
  <input type="button" value="테스트">
  <hr>
</form>
</body>
</html>
```





## ■ HTML Review

기본구조

기본태크

입력폼태그

DOM

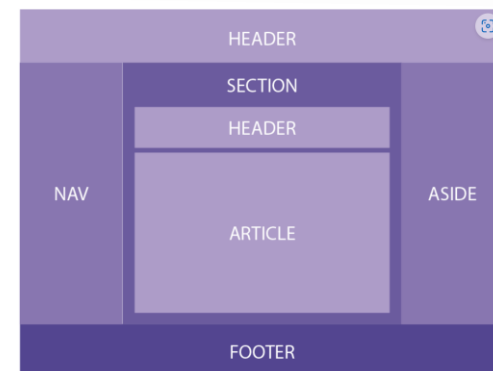
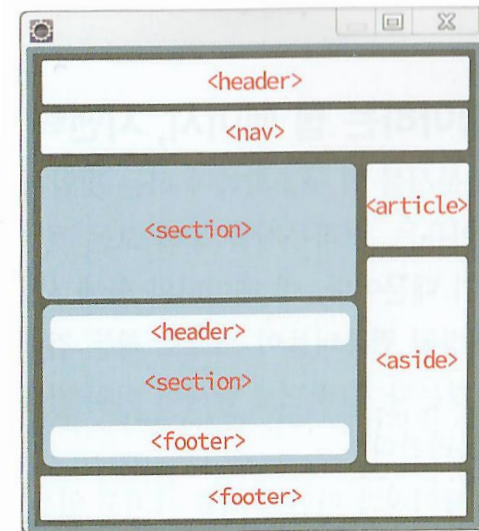
시멘틱태그

### ■ HTML 문서의 시멘틱구조

#### ☑ 시맨틱 태크

- <div>로 문서구조를 표현했으나,  
좀더 체계적으로 문서구조를 설계하는 시멘틱태그
- 의미적 구조만 정의!, 외향/모양/색은 CSS로 표현!
- \* 검색엔진이 페이지를 분석하고, 유지보수에 적합함

- 페이지의 머리말: <header> ... </header>
- 하이퍼링크의 모음: <nav> ..... </nav>
- 문서의 메인 장/절: <section> ... </section>
- 문서의 보조적인 콘텐츠: <article> ... </article>
- 문서의 비보조적인 콘텐츠: <aside> ... </aside>
- 페이지의 꼬리말: <footer> ... </footer>
- 그림을 블록화: <figure> ... <img src=...> ... </figure>



## ■ CSS Review

기본개념

Selectors

Layout

Font

### ■ 웹사이트의 스타일링

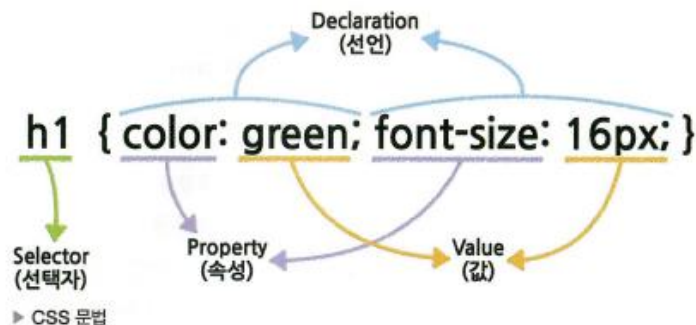
☑ 스타일링 ... 웹페이지의 전체레이아웃, 버튼색상/크기, 폰트색상/크기/폰트체 태그 엘리먼트의 스타일을 정의하고 설정하는 기능

☑ 문법

- 선택자(Selector)
- 스타일선언(Property, Value)

☑ 주요 선택자(Selector)

- Element selector
- ID selector
- Class selector
- Universal selector
- Group selector
- 상태관련 selector



## ■ CSS Review

기본개념

Selectors

Layout

Font

### ☑ 주요 선택자(Selector)

- Element selector ... HTML 태그(엘레먼트)의 속성을 스타일링
- ID selector ..... ID가 정의된 엘레먼트의 속성을 스타일링 ( # 사용 )
- Class selector ..... Class 가 정의된 HTML코드의 속성을 스타일링 ( . 사용 )
- Universal selector ... 전체 모든 엘레먼트에 대해서 속성을 스타일링
- Group selector ..... 여러 엘레먼트를 그룹으로 묶어 하나의 속성을 스타일링
- 상태관련 selector ... 엘리먼트의 상태에 따른 속성을 스타일링

```
h1 {  
  color : green;  
}
```

```
h2 {  
  color : red;  
  font-size: 16px;  
}
```

```
p {  
  color: green;  
}
```

```
<div id="myse1">  
  ...  
</div>  
  
#myse1 {  
  background-color: red;  
}
```

```
p.myse1 {  
  font-size: 30px;  
}
```

```
<span class="medium">  
  ...  
</span>  
<p class="medium">  
  ...  
</p>  
  
.medium {  
  font-size: 20px;  
}
```

## ■ CSS Review

기본개념

Selectors

Layout

Font

### ☑ 주요 선택자(Selector)

- Element selector ... HTML태크 요소의 속성을 스타일링
- ID selector ..... ID가 정의된 요소의 속성을 스타일링
- Class selector ..... Class가 정의된 HTML코드의 속성을 스타일링
- Universal selector ... 전체 모든 요소에 대해서 속성을 동일하게 스타일링
- Group selector ..... 여러 요소를 그룹으로 묶어 하나의 속성을 스타일링
- 상태관련 selector ... 요소의 상태에 따른 속성을 스타일링

```
* {  
  font-size: 16px;  
  color: blue;  
  font-family: Times;  
}
```

```
h1, h2, p {  
  color: black;  
  text-align: center;  
}
```

```
button:hover {  
  font-weight: bold;  
}  
  
input:focus {  
  color: #000000;  
}
```

## ■ CSS Review

기본개념

Selectors

Layout

Font

### ☑ Layout

- 화면에 엘리먼트를 배치하고 구성하기 위한 목적의 속성
- `display` 속성, `visibility`속성, `position`속성, 가로세로, 여백속성

```
div {  
  display: none | block | inline | flex ;  
}
```

보이지않음    블록박스형태    1줄전체 박스    컨테이너안에  
아이템을  
자유자재로 구성

```
div {  
  visibility: inherit | visible | hidden ;  
}
```

기본값  
배치    해당엘리먼트  
내용이 보임    영역을 차지하며  
내용은 보이지않음

```
div {  
  position: static | fixed | relative | absolute ;  
}
```

기본값  
배치    항상  
고정위치  
(스크롤X)    부모태크  
기준으로  
상대적 배치    HTML태크  
위치기준으로  
배치

```
div {  
  width: auto | value;  
  height: auto | value;  
  min-width: auto | value;  
  min-height: auto | value;  
  max-width: auto | value;  
  max-height: auto | value;  
}
```

### - Flexbox

다양한 Layout를 자유롭게 구성하기 위한 기능

플렉스 컨테이너



▶ 플렉스박스

## ■ CSS Review

기본개념

Selectors

Layout

Font

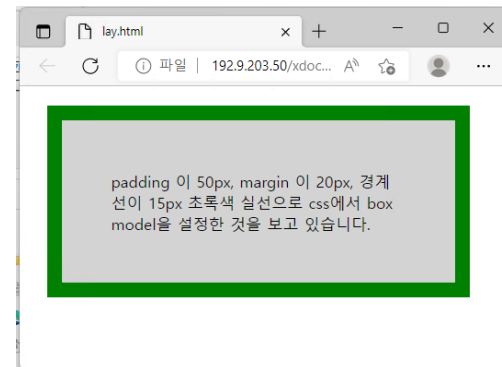
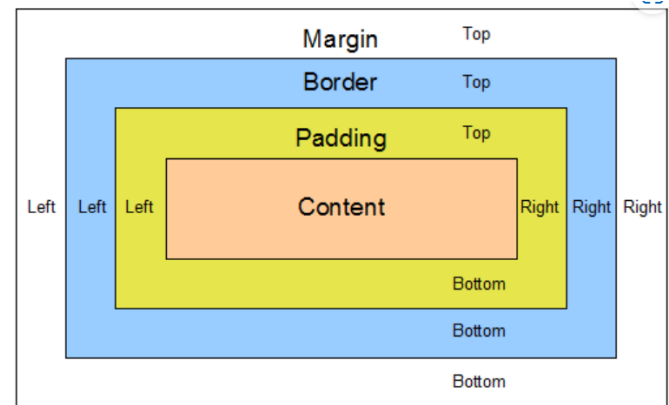
### ☑ Box model (property)

- HTML 내용(Content)를 박스형태로 출력하는 모델
- CSS에서 사용하는 대부분 property는 margin, border, padding 과 관련이 있음

```
<style>
  div {
    background-color: lightgrey;
    width: 300px;
    border: 15px solid green;
    padding: 50px;
    margin: 20px;
  }
</style>
```

margin-left: 50px  
margin-right: 90px

<div> padding 이 50px, margin 이 20px,  
경계선이 15px 초록색 실선으로 css에서  
box model로 설정한 것을 보고 있습니다.  
</div>



## ■ CSS Review

기본개념

Selectors

Layout

Font

### ☑ 폰트의 속성을 설정

- 폰트의 글꼴명, 크기, 굵기, 스타일, 색상을 정의하고 설정하기 위한 목적의 속성

```
#title {  
  font-family : "Times New Roman", Times, serif;  
  font-size: 19px;  
  font-weight: normal | bold;  
  font-style: normal | italic | oblique;  
}
```

"Times New Roman" 폰트가 없다면, Times를 선택,  
Times도 없다면, serif로 선택,...

일반글자를 기울여서 보여줌

준비된 기울인 글자를 그대로 보여줌

```
<style>  
  div {  
    background-color : lightgrey;  
    border: 3px solid black;  
    margin: 30px;  
    text-align: center;  
    text-decoration-line: underline;  
    text-decoration-color: red;  
    text-transform: uppercase;  
    letter-spacing: 5px;  
  }  
</style>
```

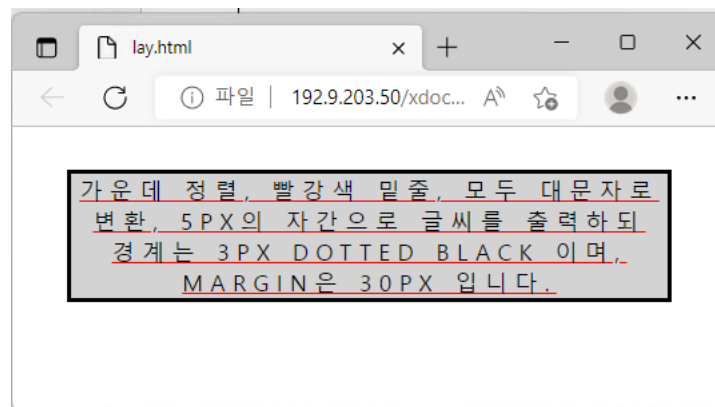
가운데 정렬

밑줄긋기

빨강색 밑줄

모두 대문자로 출력

글자간격



<div> 가운데 정렬, 빨강색 밑줄, 모두 대문자로 변환, 5px의 자간으로 글씨를 출력하되 경계는 3px dotted black이며, margin은 30px 입니다. </div>



## ■ JS Review

자료형/변수

연산

제어흐름

함수

객체

### ■ 자료형

- ☑ 숫자 <sup>number</sup> ..... 27, 4, 123, 61.98, 0.5, 0, -9.12
- ☑ 문자열 <sup>string</sup> ..... 'Hello', "Hello World", 'Oh my god', "Good!", 'This is a "father"'
- ☑ 불리언 <sup>boolean</sup> ..... true, false, !false (19 > 7, 45 > 651)

- 
- ☑ 함수 ..... 코드의 집합 (선언적 함수, 익명함수)
  - ☑ 배열 ..... 여러 종류의 자료형을 내부에 넣는 자료형, 인덱스로 구분
  - ☑ 객체 ..... 여러 종류의 자료형을 내부에 넣는 자료형, 키워드로 구분

### ■ 변수 (전역변수: **var**, 지역변수: **let**, 상수: **const**) ☞ ECMA6 에서 let, const 도입

- ☑ 변수선언 ... **var** pi;    **const** mypi=3.14;    **let** str='Hello';    **var** flag=true;
- ☑ 변수사용 ...    pi = 3.14159;    hours = 5;    str = "test A";    flag = false;

## ■ JS Review

자료형/변수

연산

제어흐름

함수

객체

### ■ 연산자

☑ 산술연산 ..... + - \* / % ++ --

☑ 문자열연산 ..... +

'Oh' + 'my god'

☑ 비교연산 ..... >= <= > < == != === !==

☑ 논리연산 ..... ! || &&

↙ ↘  
값과 자료형  
모두비교

### ■ 예

☑ `sum = sum + 5;`

☑ `code = 'initial' + 'bmlee';`

☑ `32 >= 45`

☑ `console.log(`39 == "39" : ${39 === "39"}`);` // `39 == "39" : false`

템플릿문자열 (문자열내부에 \${ ...} 를 사용할 수 있음. 즉, ... 를 수행함)

## ■ JS Review

자료형/변수

연산

제어흐름

함수

객체

### ■ 조건문 (if, switch)

```
if (result > 10) {  
    console.log('10초과');  
}
```

```
if (result > 10) {  
    console.log('10초과');  
}  
else {  
    console.log('10이하');  
}
```

```
if (result > 10) {  
    if (flag) {  
        console.log('10초과');  
    }  
    else {  
        console.log("good");  
    }  
}  
else {  
    console.log('10이하');  
}
```

```
if (3 <= month && month <= 5) {  
    console.log('봄');  
}  
else if (6 <= month && month <= 8) {  
    console.log('여름');  
}  
else if (9 <= month && month <= 11) {  
    console.log('가을');  
}  
else {  
    console.log('겨울');  
}
```

```
switch (result) {  
    case 1: result++;  
            break;  
    case 2: result--;  
            break;  
    default:  
        console.log('other');  
        break;  
}
```

## ■ JS Review

자료형/변수

연산

제어흐름

함수

객체

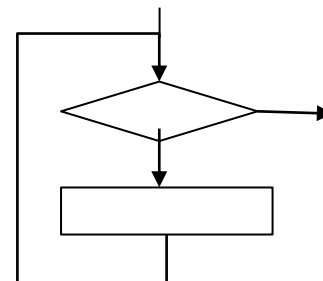
### ■ 반복문 (while)

```
let flag = true;
let sum = 10;

while (flag) {
  sum = sum + 10;
  flag = sum < 20;
}
```

```
let index = 0;
let array = ['가','나','다'];

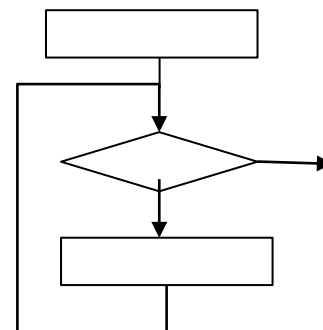
while (index < array.length) {
  console.log(array[index++]);
}
```



### ■ 반복문 (for)

```
let array = ['가', '나', '다'];

for (let index=0; index < array.length; index++) {
  console.log(array[index]);
}
```



## ■ JS Review

자료형/변수

연산

제어흐름

함수

객체

### ■ 반복문 (for in, for of)

```
let array = ['가', '나', '다'];

for (let index in array) {
  console.log(` ${index} 번째 값: ${array[index]} `); // 0번째 값: 가
}

for (let item of array) {
  console.log(item); // 가
}
```

### ■ 반복문 (배열객체의 **forEach** 함수를 이용한 반복처리)

```
let array = ['가', '나', '다'];

array.forEach( (item, index) => {
  console.log(` ${index} 번째 값: ${item} `); // 0번째 값: 가
});
```

## ■ JS Review

자료형/변수

연산

제어흐름

함수

객체

### ■ 함수선언

☑ 선언적함수 vs 익명함수 vs 화살표함수

```
function 함수이름 (a,b,c) {  
};
```

```
function (a,b,c) {  
};
```

```
(a,b,c) => {  
};
```

```
a => {  
};
```

파라미터가 1개 (예, a)뿐일 경우에는 괄호의 생략가능

```
a => a+5 = a => { return (a+5); }
```

리턴명령으로 1개일 경우

```
const yourFunction = (a,b,c) => { ... }
```

```
const myFunction = a => a+5
```

// 익명함수를 호출하려면 변수를 이용

### ■ 함수호출

```
함수이름 ();
```

```
myFunction(10);
```

## ■ JS Review

자료형/변수

연산

제어흐름

함수

객체

### ■ 매개변수 & 반환값

```
function 함수이름 ( 매개변수1, 매개변수2, ... ) {  
    return 반환값  
};
```

```
var addition = function (k, p) {  
    return k + p;  
};  
  
console.log(addition(3, 9));
```

### ■ Callback 함수 .... 매개변수로 전달되는 함수

```
function Calltentimes ( callback ) {  
    for (var c=0; c < 10; c++) {  
        callback();          // 함수호출  
    }  
};  
  
Var callback = function ( ) {  
    console.log("함수호출됨");  
}  
  
Calltentimes(callback); // 함수호출
```

Callback 함수사용

```
function Calltentimes ( callback ) {  
    for (var c=0; c < 10; c++) {  
        callback();          // 함수호출  
    }  
};  
  
Calltentimes ( function () {  
    console.log("익명으로 함수호출");  
} );
```

익명 Callback 함수사용



## ■ JS Review

자료형/변수

연산

제어흐름

함수

객체

### ■ 다른 함수표현 법

```
function 함수이름 ( 매개변수1, 매개변수2, ... ) {  
    return 반환값  
};
```

```
var addition = function (k, p) {  
    return k + p;  
};  
  
console.log(addition(3, 9));
```

```
( 매개변수1, 매개변수2, ... ) => {  
};
```

```
addition = (k, p) => {  
    return k + p;  
};  
  
console.log(addition(3, 9));
```

## ■ JS Review

자료형/변수

연산

제어흐름

함수

객체

### ■ 객체선언

```
var product = {  
  name: '비타민C',  
  type: '건강식품',  
  price: 10000,  
  stock: true  
};
```

```
var avatar = {  
  model: 100,  
  host: 'bmlee',  
  energy: 100,  
  mylocation: [277, 200, 21],  
  eat: function ( food ) {  
    if (food > 100)  
      (this.energy)++;  
  },  
  run: function ( ) {  
    (this.energy)--;  
  }  
}
```

### ■ 객체사용

또는

```
console.log( product['name'] );  
console.log( product['type'] );  
console.log( product['price'] );  
console.log( product['stock'] );
```

```
console.log( product.name );  
console.log( product.type );  
console.log( product.price );  
console.log( product.stock );
```

```
console.log(avatar.model);  
console.log(avatar.host);  
avatar.eat(300);  
avatar.run();
```



실습일지

## 강의 Q&A