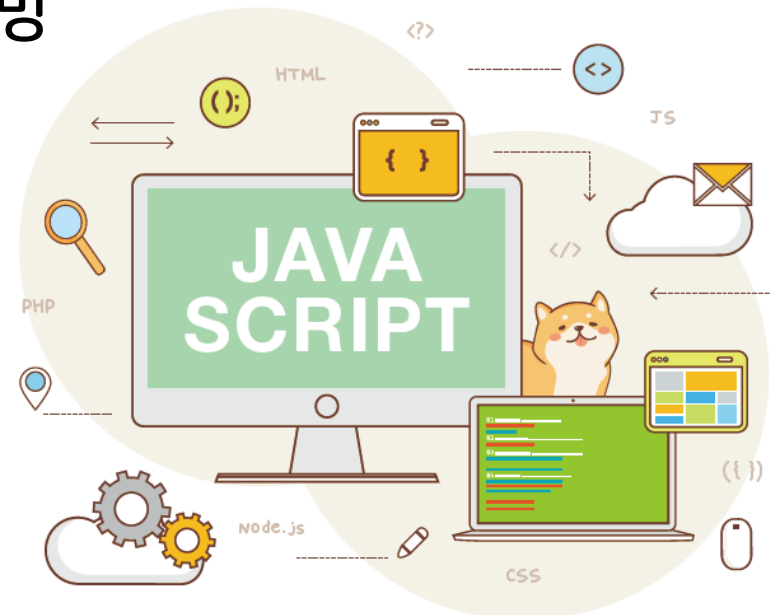


자바스크립트 기본 (Chapter4,7)

스마트헬스 케어 – 4주차

학습 내용

1. 문자열 다루기
2. 배열 다루기
3. 다중 모듈(파일)을 이용한 프로그래밍



이론 및 예제 실습

1_문자열

- 문자열은 기본적으로 상수의 속성임
- 길이 : length
- 문자단위 접근 : `charAt([인덱스])`, `charCodeAt([인덱스])` → 유니코드 값

```
var firstStr = "이것도 문자열입니다."; // 큰따옴표를 사용한 문자열
var secondStr = "이것도 문자열입니다."; // 작은따옴표를 사용한 문자열
var thirdStr = "나의 이름은 '홍길동'이야."; // 작은따옴표는 큰따옴표로 둘러싸인 문자열에만 포함
var fourthStr = '나의 이름은 "홍길동"이야.'; // 큰따옴표는 작은따옴표로 둘러싸인 문자열에만 포함
```

```
var strKor = "한글";
var strEng = "abcABC";
console.log(strKor.length); // 2
console.log(strEng.length); // 6
```

```
str = "abcDEFabc";
console.log(str.charAt(0)); // a
```

1_문자열 - 찾기

- indexOf([찾을 문자열], [시작인덱스:생략하면 0부터]); → 찾은 시작 인덱스를 알려 줌 (못 찾으면 -1)
- replace(/[찾을 문자열]/gi, [바꿀문자열]) → 문자열을 찾아 바꿈
- replaceAll([찾을 문자열], [바꿀문자열]) → 문자열을 찾아 바꿈

```
let str = "abcDEFabc";
```

```
console.log(str.indexOf("abc")); // 0 -> 자바스크립트에서 인덱스는 0부터 시작함.
```

```
console.log(str.indexOf("abcd")); // -1 -> 문자열을 비교할 때 문자의 대소문자를 구분함.
```

```
console.log(str.indexOf("abc", 3)); // 6 -> 인덱스 3부터 'abc'를 찾기 시작함.
```

```
console.log(str.replace(/abc/gi, "ddd")); // "abc"로 하면 1개만 바꿈
```

```
console.log(str.replaceAll("abc", "ddd")); // 최신 브라우저에서 적용
```

1_문자열 - 문자열 추출

- 인덱스와 개수로 지정 : `substr([시작인덱스], [추출문자수])`
- 구분자로 문자열 나누기 : `split([구분자])` → String 배열로 리턴

```
let str = "abcDEFabc";  
console.log(str.substr(2, 4)); // cDEF      -> 시작인덱스, 추출문자수
```

```
let str2 = "자바스크립트는 너무 멋져요! 그리고 유용해요.";  
console.log(str2.split(" ")); // 띄어쓰기(" ")를 기준으로 나눔.  
console.log(str2.split("!")); // 느낌표("!")를 기준으로 나눔.
```

1_문자열 - 기타

- 문자열 합치기 : `concat([추가문자열])`
- 대문자, 소문자 변경 : `toUpperCase()`, `toLowerCase()`

```
let str = "자바스크립트";  
console.log(str.concat("는 너무 멋져요!")); // 자바스크립트는 너무 멋져요!
```

```
let str2 = "abcDEFabc";  
console.log(str2.toUpperCase());  
console.log(str2.toLowerCase());
```

2_배열 - 기본

- 배열 생성 하기

- . 배열 리터럴 대괄호([])를 사용하여 만드는 방법
- . Array() 생성자 함수로 배열을 생성하는 방법

- 배열에 값 추가 : push() 함수

```
arr.push("newItem");
```

- 배열 요소 접근 : 인덱스 사용

```
let value = arr[2];
```

- 배열 요소값 변경

```
arr[2] = 100 ;
```

- 배열 요소 삭제 하기

- . 맨 뒤에 있는 것 삭제

```
arr.pop();
```

- . 지정한 요소 삭제 : delete

```
delete arr[2];
```

- 요소 전체 다루기 : for-in, for-of 반복문 이용

2_배열 - 주요 함수

```
var arr = ['zero', 'one', 'two'];
```

```
var arr2 = new Array();
```

```
arr2.push('zero');
```

```
arr2.push('one');
```

```
arr2.push('two');
```

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

```
arr2[2] = "1";
```

```
for (let i of arr) {  
    console.log(i);  
}
```

```
for (let i in arr) {  
    console.log(i, arr[i]);  
}
```

2_배열 - 정렬 함수 / 찾기 함수

- 순서 정렬하기 : sort()

. 기본은 오름차순. 내림차순으로 하려면 sort() 후 reverse() 해 준다.

```
let arrayA = ["고구마", "감자", "바나나"];  
console.log(arrayA);  
arrayA.sort(); // 기본 오름차순  
console.log(arrayA);  
arrayA.sort().reverse(); // 내림차순  
console.log(arrayA);
```

- 요소 찾기 : indexOf ([찾을 요소]);

→ 찾은 시작 인덱스를 알려 줌 (못 찾으면 -1)

```
let arrayB = ["고구마", "감자", "바나나"];  
console.log(arrayB.indexOf("고구마"));  
console.log(arrayB.indexOf("수박"));
```

2_배열 - 주요 함수

- 주요 함수들

. slice([시작 인덱스], [종료 인덱스]) : 시작 인덱스부터 종료 인덱스 바로 앞까지의 배열 요소를 추출.

인수로 종료 인덱스가 전달되지 않으면 마지막 배열 요소까지 모두 추출

. concat([배열1], [배열2]) : 인수로 전달받은 배열들을 합쳐서 만든 새로운 배열을 반환합니다.

. toString() : 배열을 문자열로 반환. 요소간에는 쉼표로 구분

```
var arr = [1, true, "JavaScript", "자바스크립트"];
```

```
console.log(arr.slice(1, 3));
```

```
console.log(arr.slice(1));
```

```
console.log(arr.concat([2], [3, 4]));
```

```
console.log(arr.concat("다섯", [6, 7]));
```

```
console.log(arr.toString());
```

3_다중 파일 사용 프로그래밍

- 1) 자바스크립트 코드가 들어간 파일은 [모듈명].js로 저장한다.
- 2) 자바스크립트 코드를 사용하려는 html 파일에서 <script> 태그로 자바스크립트 파일을 불러 온다
형식 : <script src="[JS파일명]"></script>

test.js

```
var num = 100;

function add100(value){
  return value + 100;
}
```

ex3.html

```
<script src="test.js"></script>
<script>
  console.log(num);

  let newNum = 200;
  console.log(add100(newNum));
</script>
```

응용 예제

실습-1

실습 문제 : 국어 성적 데이터를 분리하여 총점 및 평균을 구하여 출력 하시오.
(String의 split 함수 이용)

```
let kor_score = "100,67,50,88,96,78,90,60,85,90";  
let total = 0;  
let average = 0;
```

```
console.log("총점은 ", total);  
console.log("평균은 ", average);
```

실습-2

실습 문제 : 0~9사이 임의의 정수 2개를 생성한 후 덧셈 문제를 출력하고
값을 입력 받아 정답인지 출력해 주는 프로그램 (덧셈 문제 내기)

- 0~9사이 임의의 정수를 생성하는 코드

```
Math.floor(Math.random() * 10)
```

실습-3

실습 문제 : 0~9 임의의 수 100개를 생성해서 각 숫자가 몇 개씩 생성되었는지 카운트

- 0~9사이 임의의 정수를 생성하는 코드

```
Math.floor(Math.random() * 10)
```

연속 완성 프로젝트

숫자야구 게임

1. 0~9 사이의 임의의 숫자 3개를 생성
2. 사용자 입력으로 3개의 수를 입력 받음
3. 임의로 생성한 3개의 수와 사용자 입력 받은 수를 비교
 - 사용자 입력을 받은 숫자가 임의로 생성한 3개의 수에 포함될 때
 - . 값과 숫자 순서가 같으면 스트라이크 판정
 - . 값은 같지만 숫자 순서가 다르면 볼 판정
4. 3에서 판정한 결과를 출력 (스트라이크 수, 볼 수)

숫자야구 게임

```
let rand_num = [];  
let value = 0;  
for (let i = 0; i < 3; i++) {  
    value = Math.floor(Math.random() * 10); // 0~9  
    rand_num.push(value);  
}  
console.log(rand_num);  
let user_input = [];  
for (let i = 0; i < 3; i++) {  
    value = Number(prompt(i + 1 + "번째 숫자를 입력하세요 (0~9)", 1));  
    user_input.push(value);  
}  
let strike_num = 0;  
let ball_num = 0;  
let match_index = -1;  
for (i in user_input) {  
    match_index = rand_num.indexOf(user_input[i]);  
    if (match_index !== -1) {  
        console.log(" >> " + match_index, i);  
        if (match_index === i) {  
            strike_num++;  
        } else {  
            ball_num++;  
        }  
    }  
}  
console.log("판정 -- " + strike_num + " 스트라이크 , " + ball_num + " 볼 ");  
alert("판정 -- " + strike_num + " 스트라이크 , " + ball_num + " 볼 ");
```