웹을 움직이는 근육,
JavaScript란 무엇인가?

스마트인재개발원 선영표연구원

배열(Array)

: 같은 타입의 여러 변수를 하나의 묶음으로 다루는 것

배열(Array)

: 같은 타입의 여러 변수를 하나의 묶음으로 다루는 것

** 인덱스!

배열

JAVA 배열

동일한 데이터 자료형만 저장 가능

배열의 크기가 **고정적**

정해진 배열의 크기만큼만
데이터 추가 가능

vs

JS 배열

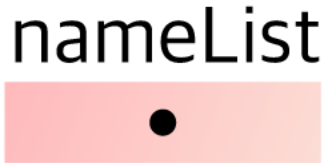
다양한 데이터 자료형 저장 가능

배열의 크기가 **가변적**

정해진 크기를 넘어서 데이터 추가 시
자동으로 저장 공간 할당

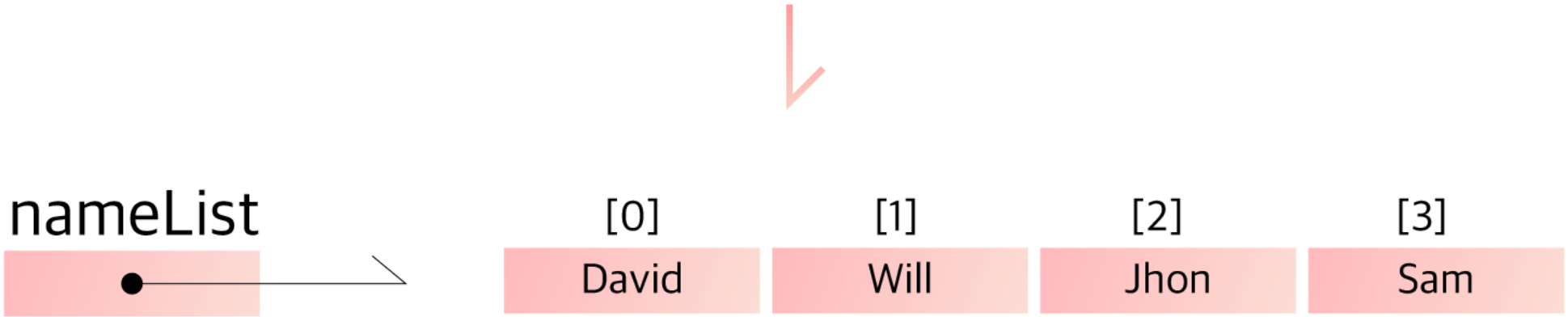
배열의 선언

```
let nameList = [];
let nameList = new Array();
```

배열의 생성

```
let nameList = [“David”,“Will”,“Jhon”,“Sam”];
let nameList = new Array(“David”,“Will”,“Jhon”,“Sam”);
```



배열에 접근

1) 반드시 배열을 **생성한 후 접근**해야한다

```
let nameList;
nameList[2] = 3;
```

```
Uncaught TypeError: Cannot set property '0' of undefined
```

-> nameList가 정의되어있지 않으므로 오류 발생!

2) 배열의 변수명과 [] 사이에 원소의 **인덱스**를 적어서 접근한다

3) 배열의 인덱스는 **0부터** 시작한다

배열실습

자바	안드로이드	총합	평균
87	81	168	84

함수(Function)

: **특정 기능**을 수행하는 소스 코드를 하나로 묶어
필요할 때마다 호출하여 사용하기 위한 구조

함수(Function)

: **특정 기능**을 수행하는 소스 코드를 하나로 묶어
필요할 때마다 호출하여 사용하기 위한 구조

** 재사용성!

내장함수

"Hello World!"

alert

함수의 사용목적?

1) 어떠한 실행코드를 묶어서 실행하기 위함

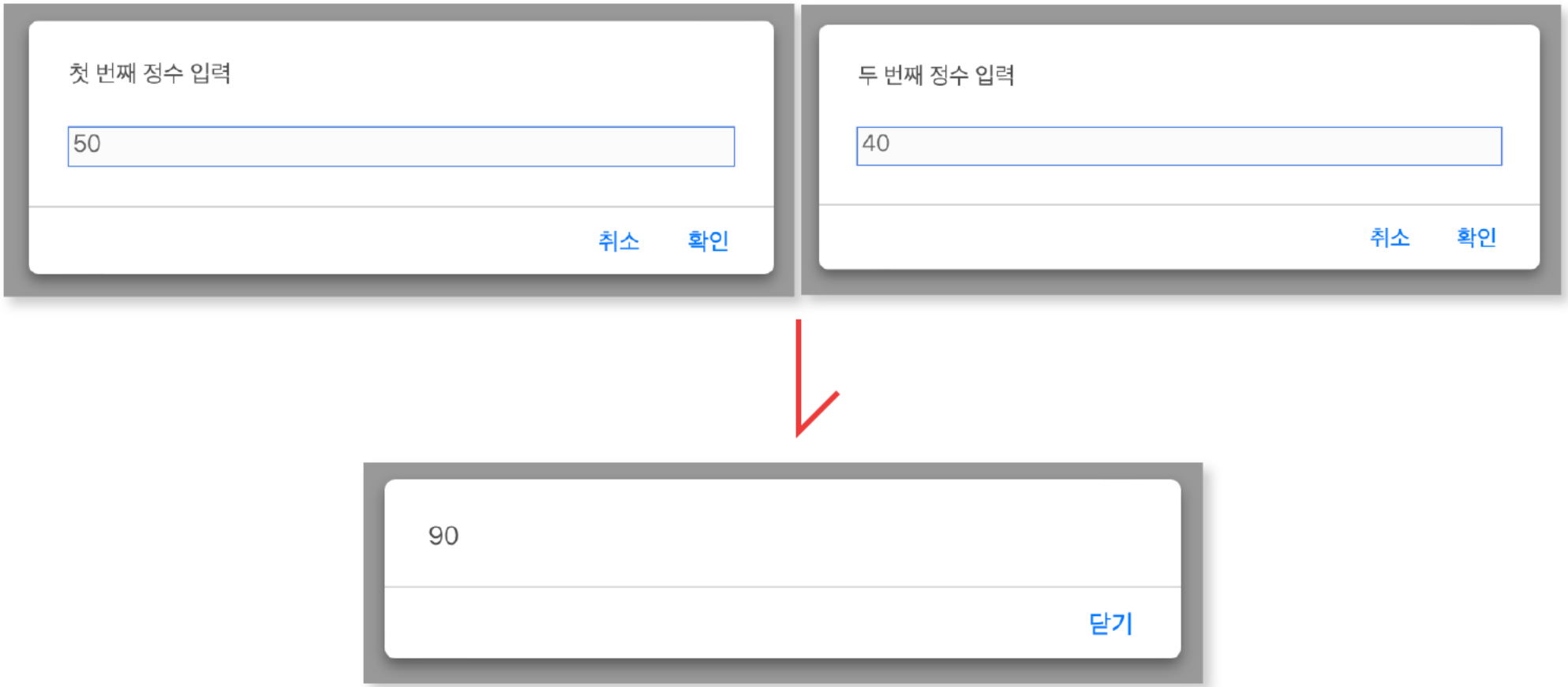
2) 중복되는 코드 최소화

3) 실행코드 블록화 >> 코드 조각화

함수의 구조

```
<script type="text/javascipt">
        function 함수명 (){
            로직 & 기능구현을 위한 코드작성
        }
        함수명(); //함수호출
</script>
```

두 정수를 입력받아 합을 구하는 addNumber() 함수를 작성하시오.



Javascript 함수의 특징

1. 매개변수와 입력값의 데이터 타입이 동일한지 검사하지 않는다.

why? **데이터 타입을 지정하지 않기 때문**

```
<script>
        var num1 = 1;
        var num2="2";

        addNum(num1,num2);

        function addNum (n1, n2){
                alert(n1+n2);
        }
</script>
```

Javascript 함수의 특징

2. 매개변수와 입력값의 개수가 같은지 확인하지 않는다.

why? 내부적으로 **arguments 객체가 호출되어 인자들을 배열 형태로 저장한다.**

```
<script>
        var num1 = 1;
        var num2 = 2;

        function addNum (n1, n2){ alert(n1+n2); }

        addNum(1,2);
        addNum(1,2,3);
</script>
```

Javascript 함수의 특징

2. 매개변수와 입력값의 개수가 같은지 확인하지 않는다.

why? 내부적으로 **arguments 객체가 호출되어 인자들을 배열 형태로 저장한다.**

```
<script>
        var num1 = 1;
        var num2 = 2;

        function addNum (n1, n2){ alert(n1+n2); }

        addNum(1,2);            => 1 2
        addNum(1,2,3);          => 1 2
</script>
```

Javascript 함수의 특징

3. 입력값의 개수가 매개 변수의 개수보다 적다면 매개 변수의 값은 undefined로 설정된다.

```
<script>
        var num1 = 1;
        var num2 = 2;

        function addNum (n1, n2){ alert(n1+n2); }

        addNum();
        addNum(1);
</script>
```

Javascript 함수의 특징

3. 입력값의 개수가 매개 변수의 개수보다 적다면 매개 변수의 값은 undefined로 설정된다.

```
<script>
        var num1 = 1;
        var num2 = 2;

        function addNum (n1, n2){ alert(n1+n2); }

        addNum();          => undefined undefined
        addNum(1);         => 1 undefined
</script>
```

화살표함수(Arrow Function)

ES6 문법으로 익명 함수 선언 참조 방식으로 표현

```
var addNum2 = (num,num2) => {
    return num+num2
}
```

객체(Object)

: 여러 속성을 하나의 변수에 저장할 수 있도록 해주는 데이터 타입

데이터(속성)과 데이터에 관련되는 동작(절차,방법,기능)을 모두 포함한 개념

**** key & value 로 접근!**

객체의 기본구조

```
<script>
let 객체명 = {
    속성명1 : 값1,
    속성명2 : 값2,
    속성명3 : 함수(){
                //기능구현
    }
            ...
};
</script>
```

** 객체는 **속성(property)**과 **기능(method)**으로 구성되어있다.
** 객체 내에는 **기본데이터타입, Array,Object** 등 데이터를 담을 수 있다.
** 객체 내 데이터를 접근하는 방법은 **마침표(.)**를 이용하는 것이다.

객체의 생성방법

```
<script>
//객체 생성
let 객체명 = {};

//속성추가
객체명.속성1 = 값1;
객체명.속성2 = 값2;
객체명.속성3 = 함수(){
            //기능구현
}
</script>
```

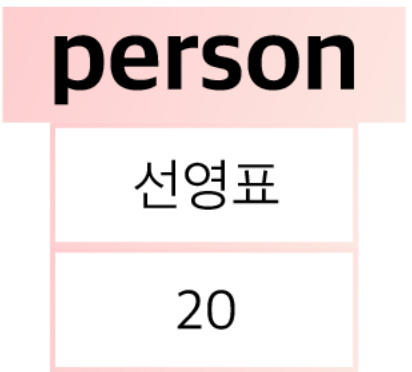
```
<script>
let 객체명 = {
    속성명1 : 값1,
    속성명2 : 값2,
    속성명3 : 함수(){
                //기능구현
    }
            ...
};
</script>
```

객체 실습

이름이 ‘선영표’이고 나이는 20살인 person객체를 생성하시오.

```
<script>
//object 객체 생성
let person = {};

//속성(property) 추가
//객체명.프로퍼티
person.name = '선영표';
person.age = 20;
</script>
```



Javascript엔진 내 메모리 할당

DOM

: Document Object Model

DOM

: Document Object Model

문서 html

DOM

: Document Object Model

문서 html 객체

DOM

: Document Object Model

문서 html 객체 모델링

WEB의 구성

HTML CSS JS

WEB의 구성

HTML CSS JS

서로 다른 언어가 소통할 수 있도록 쪼개서 **'객체화'** 시켜주는 것!

HTML요소지만 **JavaScript**를 이용할 수 있도록 해준다

HTML문서 읽는 순서

Web Browser 시작

사용자가 웹페이지 방문

HTML문서 읽는 순서

Web Browser 시작

사용자가 웹페이지 방문

웹문서 읽기

HTML문서 읽는 순서

Web Browser 시작

사용자가 웹페이지 방문

웹문서 읽기

DOM

DOM생성

HTML 문서를 전부
객체 형태로 변환
(JS로 접근 가능)

HTML문서 읽는 순서

Web Browser 시작

사용자가 웹페이지 방문

웹문서 읽기

DOM

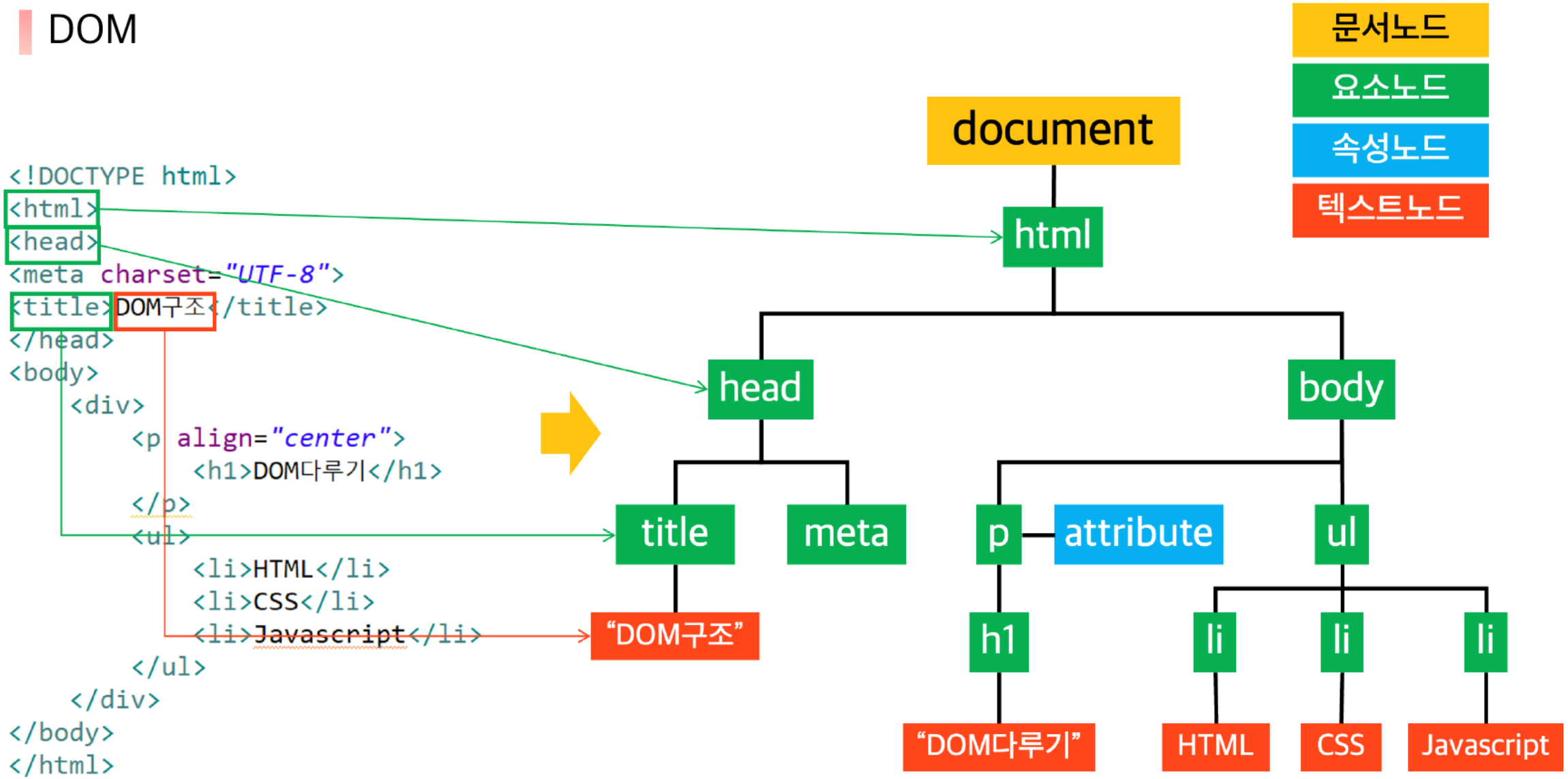
DOM생성

HTML 문서를 전부
객체 형태로 변환
(JS로 접근 가능)

페이지 로딩완료

스타일 (CSS) 적용
화면 최종표시 (렌더링)

DOM


문서노드
요소노드
속성노드
텍스트노드
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>DOM구조</title>
</head>
<body>

<h1>DOM다루기</h1>

HTML
CSS
Javascript

</body>
</html>
document
html
head
body
title
meta
p
attribute
ul
"DOM구조"
h1
li
li
li
"DOM다루기"
HTML
CSS
Javascript

HTMLElement

: 모든 종류의 HTML요소를 나타내는 인터페이스

getElement 메소드를 통해서 원하는 객체를 조회
조회된 객체들을 대상으로 구체적인 작업 처리

getElementById()

: 접근하고자 하는 HTML태그의 id값을 이용해 HTMLElement객체 조회

DOM

접근하고자 하는 요소의 id 입력

getElementById()

: 접근하고자 하는 HTML태그의 id값을 이용해 HTMLElement객체 조회