

# JAVA 웹 개발자 양성과정 Javascript

## 3강 - 데이터 타입, 연산자

By SoonGu Hong



# JAVA 웹 개발자 양성과정

## Javascript

### 1. 데이터 타입(data type)

## \* 데이터 타입

- 데이터 타입이란 **값의 종류**를 말합니다.
- 자바스크립트의 모든 값은 데이터 타입을 가지고 있습니다.
- 자바스크립트(ES6)는 7개의 데이터 타입을 제공합니다.

구분	데이터 타입	설명
기본 타입	숫자 타입(number)	숫자. 정수, 실수 구분 안함
	문자열 타입(string)	텍스트 문자열
	논리 타입(boolean)	논리적 참(true)과 거짓(false)
	undefined 타입	값을 할당하지 않은 변수에 암묵적으로 할당되는 값
	null 타입	값이 의도적으로 없다는 것을 명시할 때 사용하는 값
	심벌 타입(symbol)	ES6에서 추가된 타입
객체 타입		객체, 함수, 배열 등

## \* 숫자 타입(number)

- 자바스크립트는 C언어나 JAVA와는 달리 정수와 실수 타입을 구분하지 않습니다.
- 또한 2진수, 8진수, 16진수를 저장해도 해당 값을 참조할 경우 모두 10진수로 해석됩니다.

```
//10진수 66의 다양한 진법표현  
var bin = 0b01000010; //2진수  
var oct = 0o102; //8진수  
var hex = 0x42; //16진수
```

```
//숫자 타입은 모두 실수로 처리함  
console.log(10 === 10.0); //true
```

- NaN(not-a-number): 산술 연산 불가값.  
ex) 10 \* "hello"
- NaN은 대소문자를 반드시 이렇게 써야함. (nan, Nan, NAN 불가)

## \* 문자열 타입(string)

- 문자열 타입은 텍스트 데이터를 나타내는 데 사용합니다.
- 문자열 표기법은 홑따옴표(''), 겹따옴표(""), 백틱(`)으로 텍스트를 감쌉니다.
- 자바스크립트에서 가장 일반적인 표기법은 홑따옴표입니다.

```
var greeting;  
greeting = 'hello';      //홑따옴표  
greeting = "안녕하세요"; //겹따옴표  
greeting = `こんにちは`; //백틱
```

## \* 문자열 타입(string) - 탈출 문자

Escape character	Description of Action	Code Used in Programming
\a	Alert sound. A beep is generated by the computer on execution	"\a"
\b	Backspace.	"\b"
\f	Form feed.	"\f"
\n	New line. Shifts the cursor to new line. Words on the right of "\n" go to next line	"\n"
\r	Carriage return. Positions the cursor to the beginning of current line.	"\r"
\t	Horizontal tab, it moves the cursor by a number of spaces or to next tab stop	"\t"
\v	Vertical tab.	"\v"
\\	Backslash. Displays a black slash character (\).	"\\\""
\'	Displays a single-quote character(').	"\''"
\"	Displays a double-quote character("").	"\\""
\?	Displays question mark (?).	"\?"
\0	Null character. Marks the end of string.	"\0"

출처: <http://ecomputernotes.com/what-is-c/basic-of-c-programming/escape-sequences-in-c>

## \* 문자열 타입(string) - 템플릿 리터럴

- 자바스크립트 ES6부터 도입된 템플릿 리터럴은 문자열 처리에 편리한 여러 기능들을 지원합니다.
  - 템플릿 리터럴을 사용하려면 반드시 문자열을 백틱(`)으로만 감싸야 합니다.
- 

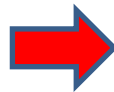
### < ES5: 템플릿 리터럴 도입 전 >

```
var template = '<ul>\n\t<li>\n\t\t<a href="#">Home</a>\n\t</li>\n</ul>';
```

### < ES6+: 템플릿 리터럴 도입 후 >

```
var template = `


  <li>
    <a href="#">Home</a>
  </li>
</ul>`;
```



```
<ul>
  <li>
    <a href="#">Home</a>
  </li>
</ul>
```

## \* 문자열 타입(string) - 템플릿 리터럴

- 템플릿 리터럴을 사용하면 변수와 문자열 결합을 간편하게 처리할 수 있습니다.
- 

```
var month = 4;  
var day = 5;  
var anniversary = '식목일';
```

< ES5: 템플릿 리터럴 도입 전 >

```
console.log(month + '월 ' + day + '일은 ' + anniversary + "입니다.");
```

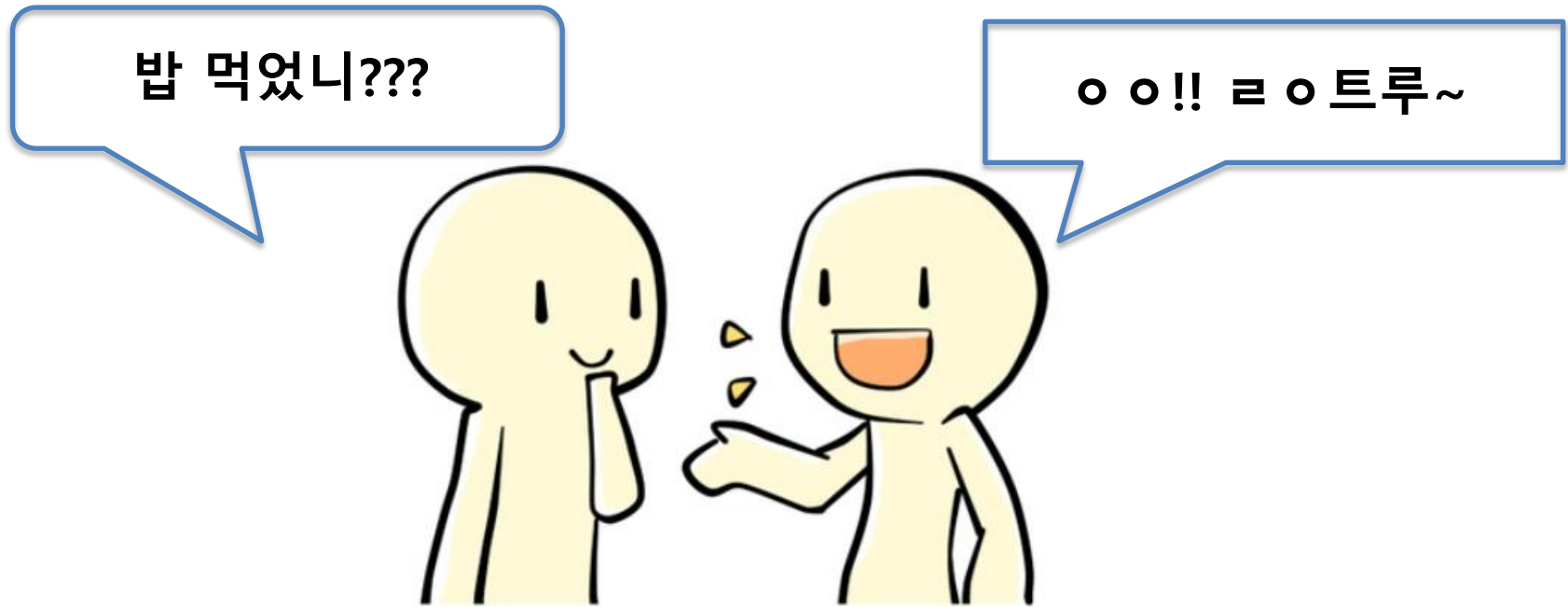
< ES6+: 템플릿 리터럴 도입 후 >

```
console.log(`${month}월 ${day}일은 ${anniversary}입니다.`);
```

4월 5일은 식목일입니다.



## \* 논리 타입(boolean)



우리는 컴퓨터에게 **논리의 참, 거짓**을 표현하게 해야 할 때가 있습니다. 이를테면 크기의 대소비교 같은 경우가 있겠죠? 그럴 때 컴퓨터가 답변하는 논리값의 형태가 **boolean**입니다.

## \* undefined, null

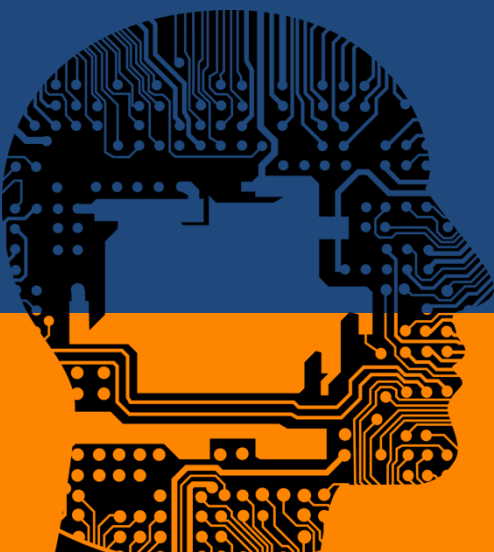
- var키워드로 선언한 변수는 암묵적으로 undefined로 초기화됩니다.
  - undefined는 개발자가 의도적으로 할당하는 값이 아닌 자바스크립트 엔진이 자동으로 넣는 값으로 보면 됩니다.
  - 변수를 참조했을 때 undefined를 확인한 경우 변수의 값이 없다는 것을 인지할 수 있습니다.
- 

- 프로그래밍 언어에서 null은 변수에 값이 없다는 것을 개발자가 의도적으로 명시할 때 사용합니다.
- 이는 이전에 할당한 값을 명시적으로 제거하는 것을 의미합니다.
- 함수가 유효한 값을 반환할 수 없는 상황에서 의도적으로 null을 반환하기도 합니다.

# JAVA 웹 개발자 양성과정

## Javascript

### 2. 연산자(operator)



## \* 연산자란?

- 연산자(operator)는 하나 이상의 표현식을 대상으로 연산을 수행하여 값을 만듭니다.
- 이 때 연산의 대상을 피연산자(operand)라고 부릅니다.
- 연산자가 연산하는 피연산자의 개수에 따라 단항, 이항, 삼항 연산자로 구분합니다.



## \* 산술 연산자

- 산술 연산자(arithmetic operator)는 피연산자를 대상으로 수학 계산을 수행해 새로운 숫자값을 만들어냅니다.
- 산술 연산이 불가능한 경우 NaN을 반환합니다.

연산자	의미	비고
+	덧셈	이항 연산자
-	뺄셈	
*	곱셈	
/	나눗셈	
%	나머지	
**	거듭제곱	이항, ES7 추가
++	1 증가	단항 연산자
--	1 감소	

## \* 할당(대입) 연산자

- 할당 연산자(assignment operator)는 우항에 있는 피연산자의 평가 결과를 좌항에 있는 변수에 할당합니다.

연산자	예시	동일 표현
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
**=	x **= 5	x = x ** 5

## \* 비교 연산자

- 비교 연산자(comparison operator)는 좌항과 우항의 피연산자를 비교한 후 그 결과를 논리(boolean)값으로 반환합니다.

연산자	의미	예시	설명
==	동등 비교	<code>x == y</code>	x와 y의 값이 같음
===	일치 비교	<code>x === y</code>	x와 y의 값과 타입이 같음
!=	부동등 비교	<code>x != y</code>	x와 y의 값이 다름
!==	불일치 비교	<code>x !== y</code>	x와 y의 값과 타입이 다름
>	대소 비교	<code>x &gt; y</code>	x가 y보다 크다
<		<code>x &lt; y</code>	x가 y보다 작다
>=		<code>x &gt;= y</code>	x가 y보다 크거나 같다
<=		<code>x &lt;= y</code>	x가 y보다 작거나 같다

## \* 논리 연산자

- 논리 연산자(logical operator)는 좌항과 우항의 피연산자를 논리연산한 결과를 도출합니다. (부정 논리 연산자의 경우 단항 연산)
- &&: 양쪽 항의 논리값이 모두 true일 경우에만 전체 결과를 true로 도출.
- ||: 양쪽 항 중에 한쪽만 true여도 전체 결과를 true로 도출

연산자	의미
&&	논리곱(AND)
	논리합(OR)
!	부정(NOT)



## \* 삼항 조건 연산자

조건식 ? 조건식이 true일 때 반환할 값 : 조건식이 false일 때 반환할 값

```
var score = 70;  
var result = score >= 60 ? '합격' : '탈락';  
  
console.log(`시험결과: ${result}`);
```



시험결과: 합격

**감사합니다**  
**THANK YOU**