



데이터 타입

- 숫자(Number), 문자열(String), 불리언(Boolean), 객체(Object), 배열(Array),
`null`, `undefined`

자바스크립트 데이터 타입 정리

자바스크립트에는 다양한 데이터 타입이 있으며, 이들 데이터 타입은 변수에 저장되는 값의 형태에 따라 구분됩니다. 이 강의에서는 기본적인 데이터 타입 5가지와 몇 가지 추가적인 데이터 타입에 대해 설명합니다.

1. 기본적인 데이터 타입

1-1. 숫자 (Number)

- 자바스크립트에서 숫자는 정수와 소수 모두 포함됩니다. 심지어 ****무한대(Infinity)****도 숫자 타입에 속합니다.

```
let age = 37;           // 정수
let height = 1.75;      // 소수
let infinityNum = Infinity; // 무한대
```

1-2. 문자열 (String)

- 문자열은 문자들의 집합이며, 큰 따옴표(" ")나 작은 따옴표(' ')를 사용하여 감싸서 표현합니다.

```
let name = "John";      // 큰 따옴표 사용
let greeting = 'Hello!'; // 작은 따옴표 사용
```

- *백틱(`)`**을 사용하면 문자열 내에서 변수를 표현할 수 있습니다. 이를 **템플릿 리터럴**이라고 합니다.

```
let age = 37;
let message = `Hello, I am ${age} years old.`; // "Hello,
I am 37 years old."
```

1-3. 불리언 (Boolean)

- 불리언은 참(True) 또는 거짓(False)만을 값으로 가질 수 있습니다.

```
let isAdult = true;
let hasLicense = false;
```

- 조건문에 주로 사용되며, 참과 거짓을 판단하는 역할을 합니다.

```
if (isAdult) {
  console.log("You are an adult.");
} else {
  console.log("You are not an adult.");
}
```

1-4. 객체 (Object)

- 객체는 키-값 쌍으로 구성된 데이터를 저장할 수 있는 복합 데이터 타입입니다.

```
let person = {
  name: "John",
  age: 34,
  isMarried: false
};
```

- 객체의 각 프로퍼티에 접근하려면 **점 표기법(dot notation)**이나 **대괄호 표기법(bracket notation)**을 사용합니다.

```
console.log(person.name); // John
console.log(person['age']); // 34
```

1-5. 배열 (Array)

- 배열은 여러 개의 값을 **순서대로 나열**할 수 있는 데이터 타입입니다. 대괄호(`[]`)를 사용하여 배열을 선언합니다.

```
let members = ["Minji", "Hanni", "Danielle"];
```

- 배열은 0부터 시작하는 **인덱스**를 가지며, 인덱스를 통해 배열의 각 요소에 접근할 수 있습니다.

```
console.log(members[0]); // Minji
console.log(members[1]); // Hanni
```

2. 추가적인 데이터 타입

2-1. `null`

- `null`은 **값이 없음**을 나타내는 특수 값입니다. 주로 변수가 **명시적으로 비어있음**을 표현할 때 사용됩니다.

```
let car = null; // 값이 없음
```

2-2. `undefined`

- `undefined`는 변수가 선언되었지만 **값이 할당되지 않은 상태**를 나타냅니다.

```
let car;
console.log(car); // undefined
```

3. 동적 타이핑 (Dynamic Typing)

자바스크립트는 **동적 타이핑(dynamic typing)** 언어입니다. 이는 변수를 선언할 때 따로 타입을 지정하지 않으며, 변수에 할당된 **값의 타입에 따라** 자동으로 결정됩니다.

```
let value = 10; // 숫자
value = "Hello"; // 문자열
value = true; // 불리언
```

4. 배열과 객체

4-1. 배열 안에 배열

- 자바스크립트에서는 배열 안에 배열이 들어갈 수 있습니다. 이를 **다차원 배열**이라고 합니다.

```
let nestedArray = [[1, 2], [3, 4], [5, 6]];
console.log(nestedArray[0]);    // [1, 2]
console.log(nestedArray[0][1]); // 2
```

4-2. 배열과 객체 혼합

- 배열 안에 객체가 들어갈 수도 있으며, 객체 안에 배열을 넣는 것도 가능합니다.

```
let team = [
  { name: "Minji", age: 19 },
  { name: "Hanni", age: 18 },
  { name: "Danielle", age: 19 }
];

console.log(team[0].name); // Minji
```

값과 객체의 차이점, **null**, **undefined** 개념 정리

1. 값과 객체의 차이

1-1. 값(Value)

- 값은 독립적인 의미를 가지는 데이터입니다. 숫자, 문자열, 불리언 등 기본 데이터 타입이 모두 값으로 취급됩니다.
- 값은 그 자체로 의미를 가지며, 다른 데이터와 결합되지 않은 상태로 존재합니다.

예시:

```
let number = 17;    // 숫자 값
let greeting = "Hello"; // 문자열 값
```

```
let isTrue = true; // 불리언 값
```

- 위의 예에서 숫자 17, 문자열 "Hello", 불리언 값 true는 각각 독립적인 의미를 가지는 값입니다.

1-2. 객체(Object)

- 객체는 **병합적인 의미**를 가지는 데이터입니다. 객체는 여러 값이나 프로퍼티를 하나의 단위로 묶어 관리할 수 있습니다. 객체는 **키-값 쌍**으로 이루어져 있습니다.
- 객체는 하나의 독립된 값이 아니라, 여러 ****속성(프로퍼티)****을 포함한 복합적인 데이터입니다.

예시:

```
let person = {  
  name: "John",  
  age: 34,  
  isMarried: false  
};
```

- 위에서 `person` 객체는 이름, 나이, 결혼 여부 등의 여러 속성을 하나로 묶어 **다중적인 의미**를 가집니다.

2. `null` 과 `undefined`

2-1. `null`

- *`null`*은 **값이 없음**을 나타내는 데이터입니다. **빈 데이터**를 명시적으로 나타내고자 할 때 사용합니다.
- `null`은 **독립적인 의미**를 가지며, 그 자체로 **빈 상태**를 의미합니다.

예시:

```
let emptyValue = null;  
console.log(emptyValue); // null 출력
```

- 이 예시에서 `emptyValue`는 값이 없음을 나타내며, **빈 값**으로 정의된 상태입니다.

2-2. `undefined`

- * `undefined` *는 변수가 선언되었으나 값이 할당되지 않은 상태를 나타냅니다. 즉, 변수가 메모리 상에 존재하지만, 아직 값을 가지고 있지 않은 상태입니다.
- 자바스크립트에서 변수 선언 후 값을 할당하지 않으면 자동으로 `undefined` 값이 할당됩니다.

예시:

```
let a;
console.log(a); // undefined 출력

a = 10;
console.log(a); // 10 출력
```

- 첫 번째 `console.log` 에서 변수 `a` 는 선언되었지만 값이 할당되지 않았기 때문에 `undefined` *가 출력됩니다. 이후 값이 할당되면 `undefined` 대신 할당된 값이 출력됩니다.

3. `null` 과 `undefined` 의 차이점

- `null` : 명시적으로 `""` 값이 없음""을 의미하는 데이터 타입입니다. 값이 없다는 것을 개발자가 의도적으로 설정합니다.
- `undefined` : 변수가 선언되었지만 값이 할당되지 않은 상태를 나타냅니다. 이는 자바스크립트가 자동으로 할당하는 값입니다.

예시:

```
let a = null; // a는 "값이 없다"로 명시
let b; // b는 선언만 되었고 값이 없음 (undefined)
console.log(a); // null 출력
console.log(b); // undefined 출력
```

마무리

- 값은 독립적인 데이터를 의미하고, 객체는 여러 값을 묶어서 관리하는 복합적인 데이터입니다.
- * `null` *은 값이 없음을 명시적으로 표현할 때 사용되며, * `undefined` *는 변수가 선언되었지만 아직 값이 할당되지 않은 상태를 나타냅니다.

이 개념들을 잘 이해하면 자바스크립트의 데이터 타입을 보다 효과적으로 사용할 수 있습니다.