



1강 - 변수와 상수

변수가 필요한 이유

변수의 선언

변수의 초기화와 할당

변수의 재할당

변수 이름 규칙

이름 관례 (Convention)

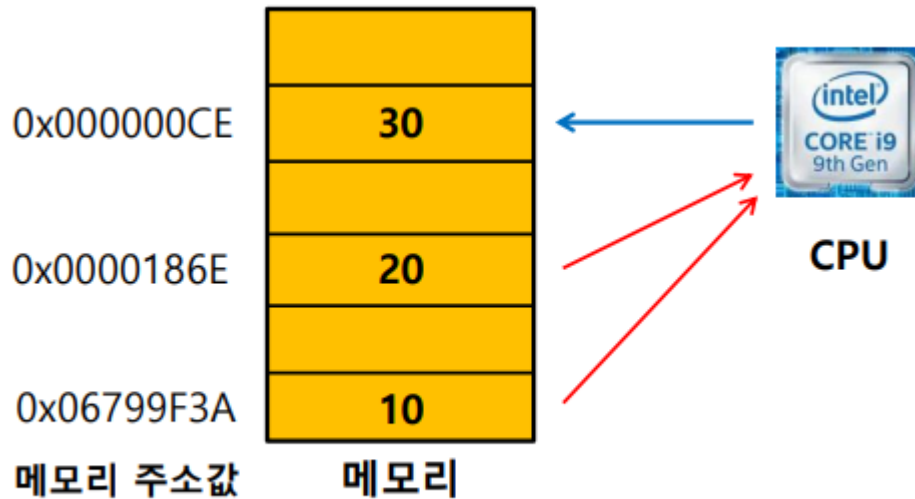
상수 (constant)

바람직한 변수명

변수가 필요한 이유

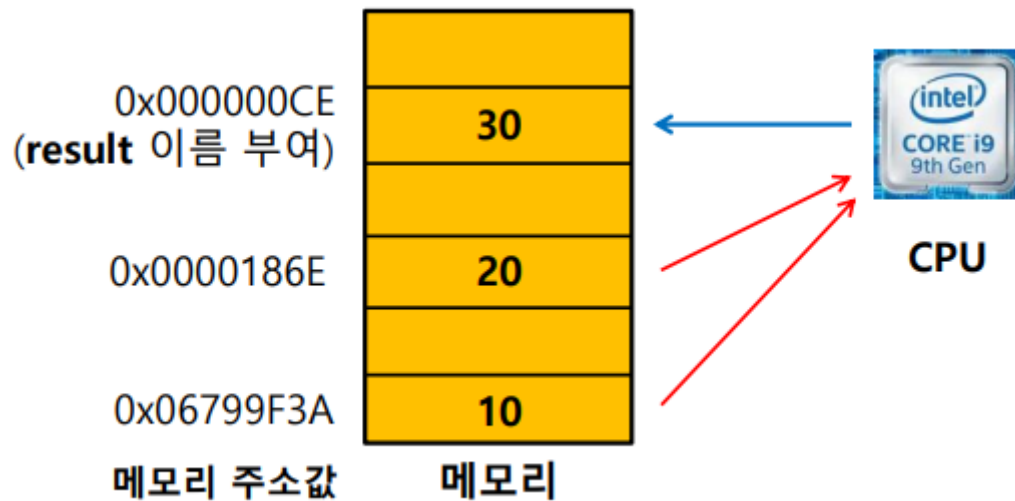
$$10 + 20$$

- 여러분은 위 수학식을 보고 뭐가 떠오르나요? 아마도 30이 떠오르실 겁니다.
- 그렇다면 컴퓨터에게 같은 질문을 한다면 어떨까요??



- 3 -

- 여기서 문제점이 하나 생깁니다. 컴퓨터는 10과 20의 합인 30을 정확히 연산했으며 메모리에 저장도 했습니다.
- 그런데 개발자가 30이란 값을 사용하기 위해서는 30이 저장된 곳의 메모리 주소값을 알아야 한다는 것이죠...
- 이 때 변수라는 개념을 사용해서 30이 저장된 공간에 사람이 식별하기 쉬운 이름을 달아 줍니다. (ex: result)
- 우리는 이제부터 result라는 이름을 통해 30이란 값을 사용할 수 있게 됩니다.



변수의 선언

- 변수를 선언한다는 것은 메모리에 데이터를 저장하는 공간을 확보하는 것을 의미합니다.
- 변수를 선언할 때는 **var**, **let**, **const** 키워드를 사용합니다. ES6에서 let, const가 도입되기 전까지는 모두 var를 사용했습니다.
- 우선은 **var**를 사용하여 변수를 선언해보고 차후에 let, const를 설명하겠습니다.

< 변수 선언 방법 >

var 변수이름;

ex) var total;

0x000000CE
(total)



메모리 주소값

메모리

변수의 초기화와 할당

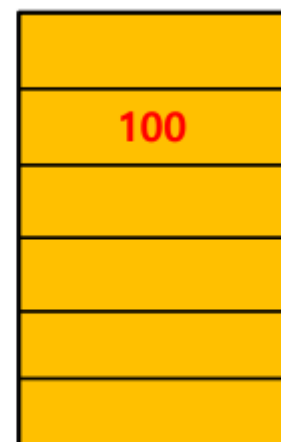
- 변수를 선언만 하고 값을 저장하지 않으면 일단 정의되지 않았다는 뜻을 가진 값인 **undefined**가 할당됩니다.
- 특정 값을 저장하고 싶으면 **할당(대입)연산자 =** 을 사용하여 값을 할당해야 합니다.
- 변수의 선언과 함께 동시에 값을 할당할 수도 있습니다.
ex) var age = 27;

< 변수 할당 방법 >

var 변수이름;
변수이름 = 값;

ex) var total;
total = 100;

0x000000CE
(total)



메모리 주소값

메모리

변수의 재할당

- 변수(variable)는 이름에서 알 수 있듯이 변경 가능한 데이터를 말합니다.
- 따라서 언제든지 저장된 값을 변경할 수 있습니다. 이를 변수의 재할당이라고 합니다.
- 이에 반대되는 개념으로 한번 할당된 값을 변경할 수 없다면 해당 개념을 **상수 (constant)**라고 합니다.

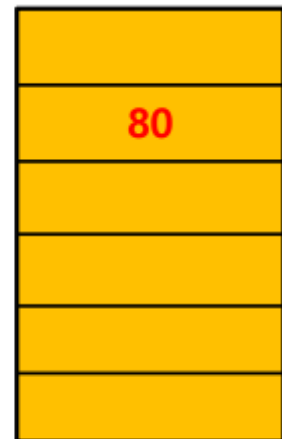
< 변수 재할당 방법 >

var 변수이름 = 값;
변수이름 = 변경값;

ex) **var total = 100;**
total = 80;



0x000000CE
(total)



메모리 주소값

메모리


변수 이름 규칙


- 식별자 이름은 **중복**을 허용하지 않고 대/소문자를 구분!
- 식별자 이름은 **숫자만으로** 구성하거나 **숫자로 시작**하면 안됨!
- 식별자 이름에 **공백**을 포함할 수 없음!
- 식별자 이름에는 **특수문자**를 사용하면 안됨(예외: **_**, **\$**)
- **키워드**(예약어)는 식별자 이름으로 사용 불가능!

ex) if, for, while, var 등...

어휘 문법 - JavaScript | MDN

이 페이지는 JavaScript의 어휘 문법(lexical grammar)을 설명합니다. ECMAScript 소스 본문은 왼쪽에서 오른쪽 방향으로 분석되고 토큰, 제어문자, 줄바꿈, 주석, 또는 공백으로 구성되는 입력

 https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Lexical_grammar

 mdn web docs

```
// 식별자: 데이터를 구분하기 위한 고유 이름(변수, 함수..)
// 1. 대소문자를 구분할 것!
var banana = '바나나';
var Banana = '빠네너';
var BaNaNa = '버네이너';

console.log(banana);
console.log(Banana);

//2. 숫자로 시작하거나 숫자만으로 만들면 안됨!
// var 800 = '메롱';
// var 70apple = '사과';
var number7 = 7;
var app99le = 99;

//3. 특수문자는 $, _ 외 사용불가
// var hi! = '안녕';
var $bye = '잘가';
var my_friend = '둘리';

//4. 띄어쓰기 불가능
// var user phone number = '01012344321';
var userphonenumber = '01099993333';

// naming convention : 이름 규칙
var user_phone_number = 1; // snake case (파이썬, SQL 선호)
var userPhoneNumber = 1; // camel case (자바, 자바스크립트 선호)
var UserPhoneNumber = 1; // pascal case (C계열)

//5. 키워드(예약어)는 이름으로 쓰지 말것!
// var var = 1;
// var if = 1;
// var break = 1;

// 식별자 이름은 구체적이고 명확하고 일관성있게 지을 것!
var a = '홍길동'; // userName
var zzzz = 'abc1234@gmail.com'; // userEmail
var fzd = '남자' // userGender

var money = 50000;
```

이름 관례 (Convention)

//네이밍 컨벤션(관례)

//카멜 케이스(camelCase)

var userPassword;



변수, 함수 이름에 사용

//스네이크 케이스(snake_case)

var user_password;

//파스칼 케이스(PascalCase)

var UserPassword;



클래스, 생성자 이름에 사용

//헝가리안 케이스(type+HungarianCase)

var strUserPassword; //str : 문자형

var intLuckyNumber; //int : 정수형

var \$button = document.getElementById('my-btn'); //\$: DOM노드

상수 (constant)

- `const` 로 선언한 변수를 '상수(constant)'라고 부릅니다. 상수는 재할당할 수 없으므로 상수를 변경하려고 하면 에러가 발생합니다.

```
const myBirthday = '18.04.1982';
```

```
myBirthday = '01.01.2001'; // error, can't reassign the constant!
```

- 기억하기 힘든 값을 변수에 할당해 별칭으로 사용하는 것은 널리 사용되는 관습입니다.
- 이런 상수는 대문자와 밑줄로 구성된 이름으로 명명합니다.

```
const COLOR_RED = "#F00";
const COLOR_GREEN = "#0F0";
const COLOR_BLUE = "#00F";
const COLOR_ORANGE = "#FF7F00";

// 색상을 고르고 싶을 때 별칭을 사용할 수 있게 되었습니다.
let color = COLOR_ORANGE;
alert(color); // #FF7F00
```

- 대문자 상수를 사용하면 좋은 점
 - `COLOR_ORANGE` 는 `"#FF7F00"` 보다 기억하기가 훨씬 쉽습니다.
 - `COLOR_ORANGE` 를 사용하면 `"#FF7F00"` 를 사용하는 것보다 오타를 낼 확률이 낮습니다.
 - `COLOR_ORANGE` 가 `#FF7F00` 보다 훨씬 유의미하므로, 코드 가독성이 증가합니다.

바람직한 변수명

- 변수명은 간결하고, 명확해야 합니다. 변수가 담고있는 것이 무엇인지 잘 설명할 수 있어야 합니다.
- 변수의 이름을 짓는 것은 프로그래밍에서 가장 중요하고 복잡한 기술 중 하나입니다. 변수명만 슬쩍 봐도 초보자가 코드를 작성했는지, 노련한 개발자가 작성했는지 알 수 있습니다.
- 변수 명명 시 참고하기 좋은 규칙
 - `userName` 이나 `shoppingCart` 처럼 사람이 읽을 수 있는 이름을 사용하세요.

- 무엇을 하고 있는지 명확히 알고 있지 않을 경우 외에는 줄임말이나 `a`, `b`, `c`와 같은 짧은 이름은 피하세요.
- 최대한 서술적이고 간결하게 명명해 주세요. `data` 와 `value` 는 나쁜 이름의 예시입니다. 이런 이름은 아무것도 설명해주지 않습니다. 코드 문맥상 변수가 가리키는 데이터나 값이 아주 명확할 때에만 이런 이름을 사용합시다.
- 자신만의 규칙이나 소속된 팀의 규칙을 따르세요. 만약 사이트 방문객을 'user'라고 부르기로 했다면, 이와 관련된 변수를 `currentVisitor` 나 `newManInTown` 이 아닌 `currentUser` 나 `newUser` 라는 이름으로 지어야 합니다.

reference:

변수와 상수



<https://ko.javascript.info/variables>



JAVASCRIPT.INFO
The Modern JavaScript Tutorial