

02

02-2. 변수와 데이터 타입

JavaScript Basic

소프트웨어 언어 학습 흐름

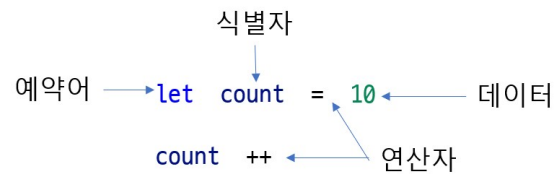
- 모든 소프트웨어 언어는 작성규칙이 있습니다. 이 작성규칙을 그 언어의 문법이라고 합니다.
- 소프트웨어 문법을 익히고 그 문법에 맞추어 프로그램을 작성해 주면 됩니다.

코드는 예약어, 식별자, 연산자, 데이터의 조합이다.

- 예약어는 소프트웨어 언어에서 지정한 단어
- 식별자는 개발자가 구분을 위해 지정한 단어
- 연산자는 특별한 연산을 위해 지정된 기호 혹은 단어
- 데이터는 값

소프트웨어 언어 학습 흐름

- 소프트웨어 언어 문법을 학습한다는 것은 그 언어에서 제공하는 예약어가 무엇이 있으며 식별자 정의 규칙은 어떻게 되고, 연산자는 무엇이 있는지등을 학습하는 것입니다.



소프트웨어 언어 학습 흐름

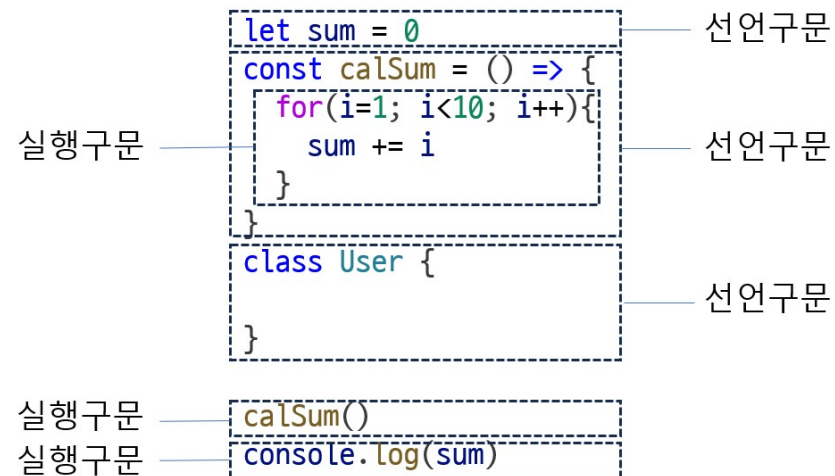
코드는 선언구문과 실행구문으로 구분된다.

- 프로그래밍 코드는 선언구문과 실행구문으로 구분할 수 있습니다.
- 선언 구문이라고 하면 “무언가 이런 것이 있다” 라고 선언하기 위해 작성되는 구문입니다.
- 변수, 함수, 클래스를 선언하기 위해서 작성되는 구문이며 흔히들 “변수를 선언해서, 함수를 선언해서, 클래스를 선언해서” 라는 표현을 많이 합니다.
- 선언구문에는 각 선언한 것을 식별하기 위해서 식별자를 지정해 줍니다.
- 변수를 선언하면서 그 변수 명을 지정하거나 함수를 선언하면서 그 함수명을 지정하게 됩니다.

소프트웨어 언어 학습 흐름

코드는 선언구문과 실행구문으로 구분된다.

- 실행구문은 이미 선언되어 있는 변수, 함수, 클래스를 이용하는 구문을 의미합니다.
- 선언된 변수에 값을 판단하거나 변경하거나, 선언된 함수를 호출하거나 아니면 선언된 클래스를 생성하는 등 무언가 선언된 것 가지고 행위가 이루어지는 구문을 실행구문이라고 합니다.



소프트웨어 언어 학습 흐름

소프트웨어 언어의 문법을 먼저 학습하고 플랫폼에서 제공하는 API 를 학습한다.



- 문법을 학습한 다음 플랫폼에서 제공되는 API 를 학습해야 합니다.
- API 라고 하면 프로그램 코드에서 이용할 수 있는 변수, 함수, 클래스등을 의미하며 이미 만들어져 제공되는 것들입니다.

소프트웨어 언어 학습 흐름

애플리케이션의 궁극적인 목적은 데이터이다. 이 데이터를 이용하기 위해 변수, 타입 등이 학습되어야 한다.

- 데이터는 애플리케이션에서 이용하는 값이라고 생각하면 되는데 이 값이 애플리케이션이 실행되는 동안 컴퓨터 메모리에 저장되어 있어야 합니다.
- 메모리에 저장된 데이터를 쉽게 식별해서 이용하기 위해서 변수를 선언하고 그 변수로 데이터를 이용합니다.
- 변수로 데이터가 이용됨으로 그 데이터가 어떤 형식의 데이터인지를 구분하기 위한 타입 이야기가 따라 나옵니다.
- 타입이란 변수에 저장된 데이터가 숫자 타입인지, 문자타입인지를 구분하는 문법이라고 보시면 됩니다.

소프트웨어 언어 학습 흐름

데이터를 적절하게 핸들링하기 위한 연산자, 제어문에 대한 학습이 필요하다.

- 애플리케이션이 실행되면서 각 데이터를 우리가 원하는 대로 조작해 주어야 합니다.
- 단순히 사칙 연산을 하거나 어느 데이터가 큰지를 판단하는 등 다양한 작업이 진행되어야 합니다.
- 이를 위해서는 연산자가 필요하며 자바스크립트에서 어떤 연산자를 제공하는지 학습이 되어야 합니다.
- 어떤 코드 부분이 반복적으로 실행되거나 조건에 만족하는 경우에만 실행되게 제어를 해주어야 합니다.
- 이런 부분을 제어문이라고 하고 자바스크립트에서 제어문을 작성하는 방법에 대한 학습이 있어야 합니다.

소프트웨어 언어 학습 흐름

하나의 업무처리를 위한 여러 라인의 코드를 묶어서 이용하기 위한 함수의 학습이 필요하다.

- 하나의 업무 처리를 위해서는 여러 라인의 코드가 필요할 수도 있습니다.
- 하나의 업무를 위한 여러 라인의 코드를 묶어서 개발하는데 그 역할을 하는 것이 함수입니다.

관련된 데이터 및 함수를 묶어서 표현하기 위한 객체에 대한 학습이 필요하다.

- 프로그램 코드를 작성하다 보면 많은 변수, 함수들이 만들어지는데 이 변수, 함수들을 관련된 것끼리 다시 묶어서 개발하고 이용합니다.
- 이를 흔히 객체라고 하며 객체를 이용하는 프로그램을 흔히 객체지향 프로그램이라고 합니다.

strict mode

- 자바스크립트 코드 첫 줄 혹은 함수 내용 첫 줄에 'use strict' 라고 쓰는 경우가 있습니다.
- 'use strict' 을 사용하는 것을 자바스크립트 '엄격모드' 라고 합니다.
- 이 엄격모드는 자바스크립트로 구현하고자 하는 코드의 알고리즘과는 관련이 없습니다.
- 우리가 작성한 자바스크립트 코드를 실행시켜 주는 자바스크립트 엔진에게 우리가 작성한 코드를 어떻게 실행시라는 일종의 명령입니다.

엄격모드 적용

```
1  "use strict";  
2  
3  //자바스크립트 구문 작성..
```

특정 함수에 엄격모드 적용

```
1  function test() {  
2      'use strict';  
3  
4      //함수 내용 작성..  
5  }
```

strict mode

- 엄격모드가 선언되지 않은 경우를 용어상으로 느슨한 모드(sloppy mode) 라고 합니다.
- 우리가 이미 살펴보았듯이 자바스크립트는 다른 소프트웨어 언어와 비교해 느슨한 문법을 지원합니다.
- 자바스크립트 프로그램을 작성하면서 엄격모드를 선언할 것인지는 개발자 선택성이지만 대부분의 자바스크립트 엔진에서 이 엄격모드를 지원하고 있으므로 엄격모드를 이용할 것을 권장하고 있습니다.

변수

변수와 데이터

- 소프트웨어 언어에서 하나의 의미있는 값을 데이터라고 하며 그 데이터를 저장하거나 변경하기 위한 구성 요소를 변수라고 합니다.



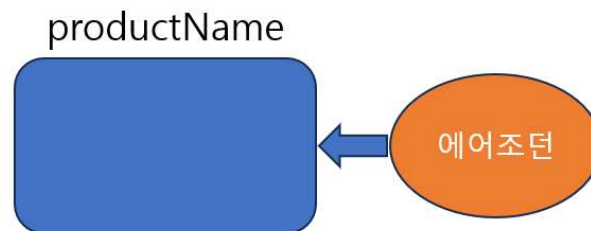
변수

변수 선언

- 자바스크립트에서 변수를 선언하기 위해서는 var, let, const 예약어 중 하나를 이용합니다.

`let` `productName` `=` `"에어조던"`

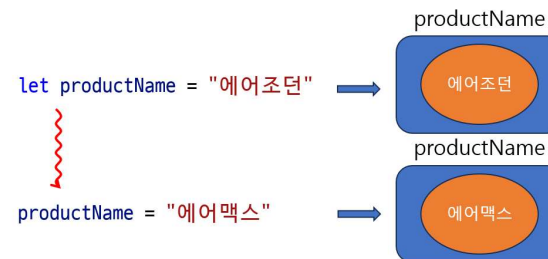
예약어 식별자 연산자 데이터



변수

변수의 데이터는 변경될 수 있다.

- 변수를 영어로 Variable 이라고 부릅니다.
- 즉 변수는 변수에 대입된 데이터가 변경될 수 있다는 의미입니다.



변수

네이밍 규칙

- 식별자는 개발자가 지정하는 임의의 단어이지만 규칙이 있습니다.
- 식별자는 문자, 숫자, 언더스코어(_), 달러기호(\$)의 조합으로 작성한다.
- 식별자의 첫글자는 문자, 언더스코어(_), 달러기호(\$)로 시작할 수 있다.
- 식별자에는 공백이 추가될 수 없다.
- 자바스크립트의 예약어는 식별자로 사용할 수 없다.

변수 네이밍 규칙	
1	<code>let 변수 = 10</code>
2	<code>let a_1\$ = 20</code>
3	<code>let _a = 30</code>
4	<code>let \$a = 40</code>

변수

네이밍 규칙

- 코드 컨벤션 측면에서 아래와 같은 규칙을 준수해 주는 것이 좋습니다.
- 두 단어가 연결되어 식별자로 사용하는 경우 camel case 로 작성한다.
- 식별자는 의미있는 단어를 이용한다.
- 변수와 함수는 소문자로 시작하고 클래스명은 대문자로 시작한다.
- 상수 변수의 경우 전체 식별자를 대문자로 작성한다.

변수

var

- var 예약어로 변수를 선언하는 방법은 자바스크립트 초기부터 제공되던 오래된 방법입니다.

`var productName = "에어조던"`

예약어 식별자 연산자 데이터

let

let 예약어로 변수를 선언하는 방법은 ECMA2015 에서 제공하는 방법입니다.

`let productName = "에어조던"`

예약어 식별자 연산자 데이터

변수

상수 변수 선언

- 상수변수는 초기 선언하면서 값을 대입하면 이후 값 변경이 불가능합니다.
- `const` 로 선언된 변수는 선언과 동시에 초기값을 대입해 주어야 합니다.

```
1 let data1
2 const data2
3
4
5
6
```

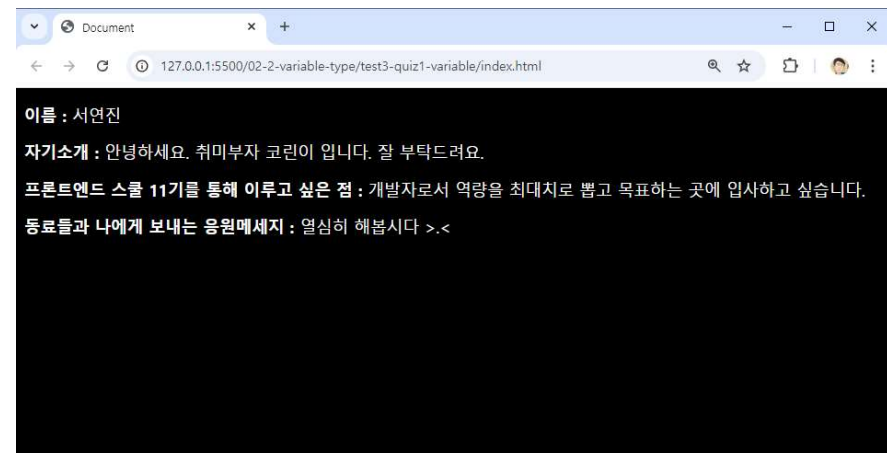
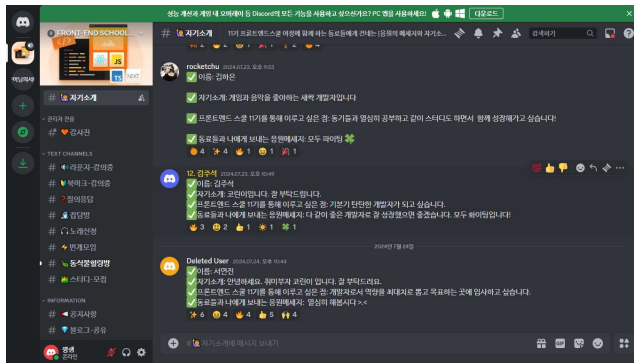
'const' declarations must be initialized. ts(1155)

`const data2: any`

[View Problem \(Alt+F8\)](#) No quick fixes available

QUIZ: 변수 선언

다음의 조건대로 코드를 작성하세요!!



- index.html, main.js 파일 생성
- discord -> 자기소개, 이곳에 작성한 자신의 소개 글을 변수를 선언해서 데이터화 하고 브라우저 화면에 위치 출력한다.
- 강사가 별도의 지시가 있을 때까지 수정한 코드를 공유하지 않습니다.
- Quiz 시간 :

데이터 타입

타입 유추

- 프로그램에서 데이터를 이용하기 위해서는 변수를 선언하고 그 변수에 데이터를 할당해서 이용합니다.
- 그런데 이런 데이터가 어떤 타입의 데이터인지를 구분해서 사용해 주어야 합니다.

10 + 10 ➡ ?

"hello" + "world" ➡ ?

데이터 타입

타입 유추

- 모든 소프트웨어 언어에서는 데이터를 타입으로 구분합니다.
- 그런데 자바스크립트에서는 개발자 코드에 타입을 명시하지는 않습니다.

자바에서 변수 선언

```
1 int no = 10;  
2 String name = "홍길동";
```

자바스크립트에서 변수 선언

```
1 let no = 10  
2 let name = "홍길동"
```

데이터 타입

타입 유추

- 자바스크립트는 데이터 타입을 위한 예약어도 없습니다.
- 하나의 변수에 다양한 타입의 데이터 대입이 가능합니다.

<code>let data = 10</code>	—————	숫자타입으로 유추
<code>data = "홍길동"</code>	—————	문자타입으로 유추
<code>data = true</code>	—————	논리타입으로 유추

- 소프트웨어 언어에서 데이터 타입이 없을 수는 없습니다.
- 데이터 타입이 없는 것이 아니라 대입되는 값을 보고 타입이 유추되는 것입니다.
- 이를 흔히 타입 유추 기법이라고 합니다.



Strong typed Language vs Weakly typed Language

- 소프트웨어 언어를 구분 짓는 용어입니다.
- 어떤 소프트웨어 언어에서 데이터 타입을 개발자 코드에서 선언해야 하고 선언된 타입의 데이터만 대입될 수 있게 하는 언어를 타입과 데이터가 강하게 결합되어 있다는 의미에서 Strong typed Language 라고 합니다.
- 대표적인 언어가 자바, C 같은 언어들입니다.
- 반대로 타입이 있기는 하지만 타입 유추 기법에 의해 타입이 판단되어 개발자 코드에서 타입을 명시하지 않아도 되는 언어를 약하게 타입과 연결된 언어라는 의미에서 Weakly typed Language 라고 합니다.

데이터 타입

- 자바스크립트에서 데이터 타입은 숫자타입, 문자타입, 논리타입 그리고 객체타입이 있습니다.

숫자타입

- 숫자 타입은 10, 20 처럼 정수 데이터와 10.1, 20.1 처럼 실수 데이터를 표현하는 타입입니다.

숫자 타입 테스트

```
1 let data1 = 10
2 let data2 = 10.0
3 console.log(`data1 type is ${typeof data1}`)//data1 type is number
4 console.log(`data2 type is ${typeof data2}`)//data2 type is number
```

- typeof 는 데이터 타입을 확인하기 위한 예약어 입니다.

데이터 타입

문자타입

- 문자 타입은 문자열 혹은 문자 하나를 표현하기 위한 타입입니다.
- 문자 타입의 데이터는 싱글쿼터(') 혹은 더블쿼터(") 로 묶여야 합니다.

문자 타입 테스트

```
1 let data1 = 10
2 let data2 = '10'
3 console.log(`data1 type is ${typeof data1}`)//data1 type is number
4 console.log(`data2 type is ${typeof data2}`)//data2 type is string
```

데이터 타입

논리타입

- 논리타입은 참/거짓 데이터를 가지는 타입입니다.
- 데이터로 true/ false 를 의미하며 프로그램 코드에서 어떤 조건을 명시할 때 많이 사용되는 타입입니다.
- 논리타입을 Boolean 타입 혹은 bool 타입이라고도 부릅니다.

논리 타입 테스트

```
1 let data1 = true
2 let data2 = 10 < 5
3 console.log(`data1 type is ${typeof data1}`)//data1 type is boolean
4 console.log(`data2 : ${data2}`)//data2 : false
```

데이터 타입

Template string

- Template string 은 Template literal 이라고도 하며 문자열 데이터를 만드는 방법을 의미합니다.
- Template string 을 만드는 목적은 동적인 결과가 추가된 문자열을 만들기 위해서 입니다.
- Template string 을 이용하기 위해서는 전체 문자열이 백틱으로 묶여야 합니다.
- 그리고 문자열내에 동적 데이터가 들어가는 부분은 `${ }` 로 묶어주어야 합니다.

`'안녕하세요 ' + name + '님, 함수 호출 결과는 ' + calSum(10) + '이고 연산 결과는 ' + (10 + 20) + '입니다.'`

Template string 으로 변경 테스트

```
1 console.log(`안녕하세요 ${name}님, 함수 호출 결과는 ${calSum(10)}이고  
   연산 결과는 ${10 + 20}입니다.`)
```

데이터 타입

undefined 와 null

- 자바스크립트의 데이터 타입과 변수에 저장되는 데이터를 이해할 때 undefined 와 null 에 대한 이해는 중요합니다.
- null 이라는 것은 자바스크립트 뿐만 아니라 모든 소프트웨어 언어에서 널리 사용되는 용어이며 개념이 모든 소프트웨어 언어에서 동일합니다.
- undefined 는 자바스크립트에만 있는 용어 혹은 개념입니다.

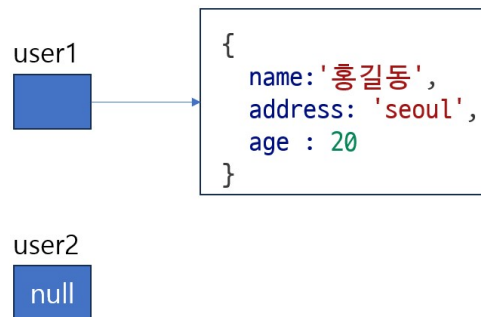
데이터 타입

null

- null 은 객체에 대입되는 값입니다.
- 그런데 무언가 의미있는 데이터를 표현하는 값이 아니라 객체가 선언되기는 했는데 아직 값이 없는 상태를 표현하는 값입니다.

null 테스트

```
1 let data1 = null
2 console.log(data1, `data1 type is ${typeof data1}`)//null 'data1 type
  is object'
```



데이터 타입

null 과 디버깅

- "Uncaught TypeError: Cannot read properties of null (reading 'style')"

```
<body>
  <script>
    let selectedNode = document.querySelector('#a1') ----- 객체가 null 인 상태 ①
    selectedNode.style.color = 'red' ----- null 인 객체를 이용하는 순간 ②
  </script>
</body>
```

- 객체가 null 인 것이 오류가 아니라 null 인 객체의 변수 혹은 함수를 이용하려고 하는 순간 에러가 발생

데이터 타입

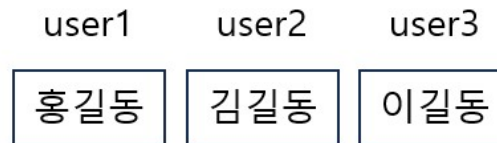
undefined

- undefined 는 데이터 타입입니다.
- 자바스크립트에서 데이터 타입은 변수에 대입되는 데이터에 의해 유추가 됩니다
- undefined 는 변수가 선언되기는 했지만 값이 대입되지 않아서 데이터 타입을 유추할 수 없는 상태의 타입을 의미합니다.
- 즉 데이터가 아니라 데이터 타입이 유추가 되지 않는 상태를 의미하는 타입입니다.

데이터 타입

컬렉션 타입 - 배열

- 배열(Array) 이라 함은 모든 소프트웨어 언어에서 제공되는 데이터 타입이며 데이터의 자료구조입니다.
- 배열을 한마디로 정의하자면 여러 개의 데이터를 표현하는 타입이라고 할 수 있습니다.



데이터 타입

컬렉션 타입 - 배열

- 배열 변수에 들어가는 데이터를 명시할 때는 대괄호([])를 이용합니다. [] 안에 콤마(,) 로 데이터를 구분해서 여러건의 데이터를 나열합니다.

```
let users = ['홍길동', '김길동', '이길동']
```

- 배열 변수의 데이터는 순서에 의해 이용됩니다.
- 이 순서를 소프트웨어 개발에서는 흔히 인덱스(index) 라고 부르며 인덱스는 0부터 시작합니다.

인덱스로 배열 이용
1 console.log(users[0], users[1], users[2])//홍길동 김길동 이길동

데이터 타입

컬렉션 타입 - 배열 사이즈, 추가, 변경, 제거

- 배열의 데이터 개수를 판단할 때는 배열의 length 변수를 이용합니다.
- length 변수는 우리가 선언하지 않아도 모든 배열에 자동으로 포함되는 변수입니다.

배열 데이터 개수 획득

```
1 let users = ['홍길동', '김길동', '이길동']
2 console.log(users.length)//3
```

- 배열에 담긴 기존의 데이터를 변경하고 싶다면 인덱스로 그 데이터의 위치를 지정하고 그곳에 변경하고자 하는 데이터를 대입하면 됩니다.

배열 데이터 변경

```
1 let users = ['홍길동', '김길동', '이길동']
2 console.log(users)//['홍길동', '김길동', '이길동']
3 //데이터 변경
4 users[1] = '박길동'
5 console.log(users)//['홍길동', '박길동', '이길동']
```

데이터 타입

컬렉션 타입 – 배열 사이즈, 추가, 변경, 제거

- 만약 기존 배열에 신규 데이터를 추가하고 싶다면 `push()` 라는 함수를 이용하면 됩니다.
- 배열에 담긴 데이터를 제거하고 싶다면 `pop()` 함수를 이용해 주면 됩니다.



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare