

Part. 1 JavaScript







목차

- 1. Getting Started
- 가. Html boilerplate
- 나. Vscode 설치
- 다. 확장프로그램 설치
- 라. 터미널 사용법, 브라우저의 콘솔창 사용법
- 2. 자바스크립트에서 자주 하는 질문
- 가 . 자바스크립트 언어는 가장 인기있는 언어이다 .
- 나 . 자바스크립트 문법의 대부분은 ,
- 다 . 자바스크립트는 어디서 실행되는가
- 라 . 자바스크립트는 ECMAScript 와 뭐가 다른가 ?
- 3. 랜더링과 파싱
- 가 . 랜더링 (Rendering)
- 나 . 파싱 (parsing)

- 4. 변수
- 가 . 식별자
- 나 . 네이밍 규칙
- 다. 변수 선언
- 라 . 상수 (Constants)
- 5. 자료형
- 가 . Primitive (Value) type
- 나. Reference type
- Nalue vs Reference Type 비교
- 6. 형변환 (타입의 변환 , 암묵적 , 명시적으로 데이타타입을 변환)
- 7. 자바스크립트는 무슨 일을 하는가



목차

- 8. 자바스크립트의 위치
- 가 . Separation of Concerns
- 나 . js 을 위한 속성 살펴보기
- 다 . script 태그안에 async / defer 속성
- 9. Operators 종류
- 10. 조건문
- 가 . if~ else, if ~ else if ~ else (원하는 만큼 사용가능)
- 나 . switch (variable)~ case: break; ~ default:
- 11. 반복문 (Loops) : for , while

for \sim in , for \sim of

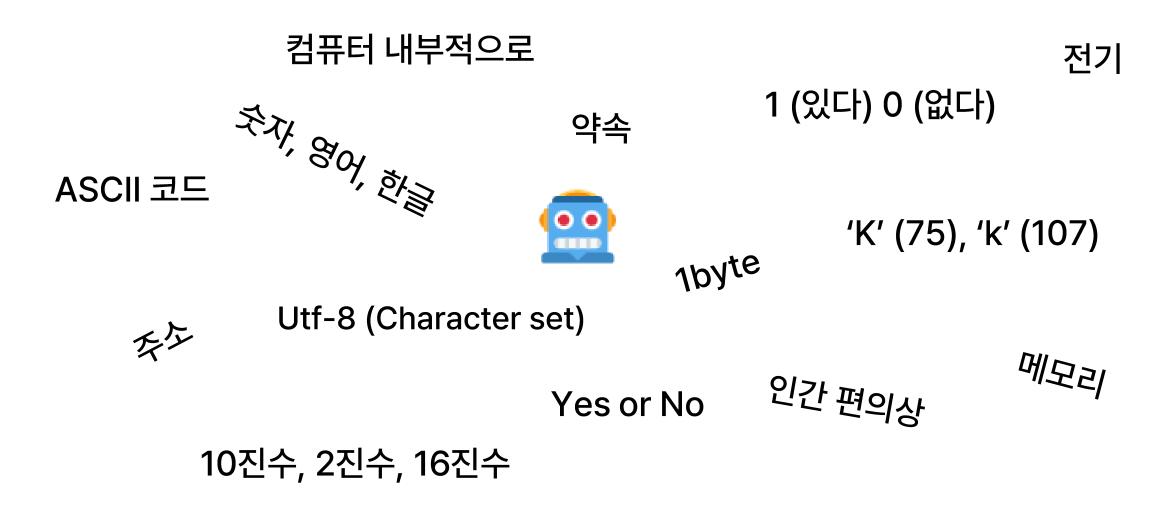
🤷iterable(이터러블) 객체

- 12. break, continue
- 13. 함수
- 가 . 함수 선언과 함수표현식
- 나. 함수의 여러가지 형태 소개
- 다. 화살표 연산자의 사용 예
- 라. 함수의 arguments, length
- 마. return문
- 14. 내장함수
- 나. 문자열 관련 메소드
- 다. Escape notation
- 15. 배열

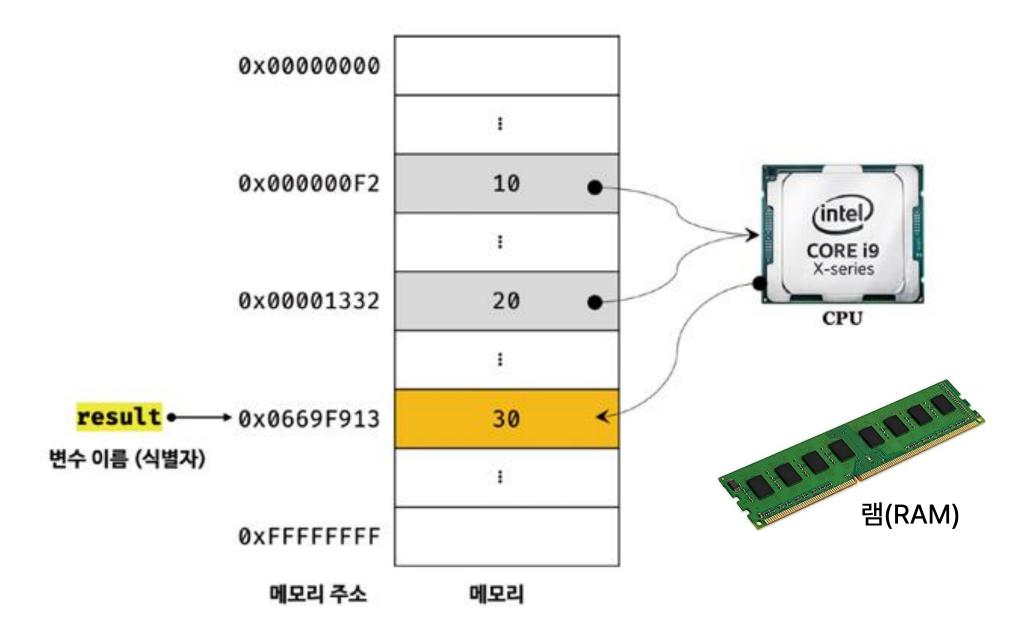
실습문제들



1. Getting Started







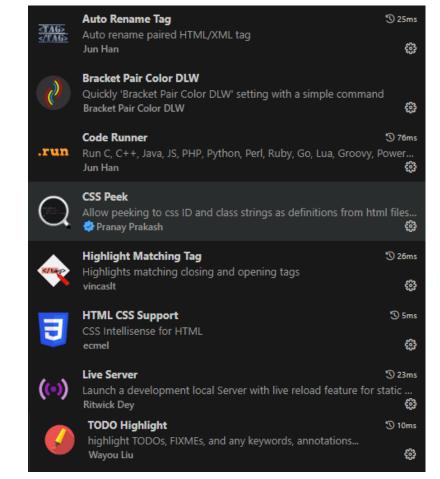


가. 에디터 설치

- html boilerplate
- 확장프로그램
- vscode 상태표시줄
- 폴더 드래그하여 열기
- 나. 실습환경

\03_javascript_basic

- 다. Vs code 의 주석처리하는 방법
 - 한줄
 - 여러줄



Carriage return Line Feed



터미널 사용법, 브라우저 콘솔창

Vs code

- 터미널 내용 지우기 : Clear
- 새로운 터미널창 : Ctrl + shift + `
- 키보드의 위,아래 화살표로 명령어 재사용
- >1+1 // 결과는 2 출력

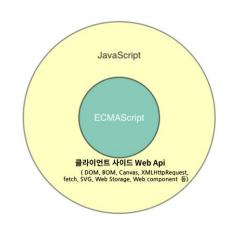
브라우저의 콘솔창

- 키보드의 위,아래 화살표로 명령어 재사용
- 콘솔창에 출력하기 위해서는 console.명령어 형식으로 입력
- 여러줄에 걸친 명령어 경우 그냥 엔터를 했을때 자동으로 다음줄로 커서가 이동한다
- >console.log(console)
- >console.clear (에러)
- >console.clear()

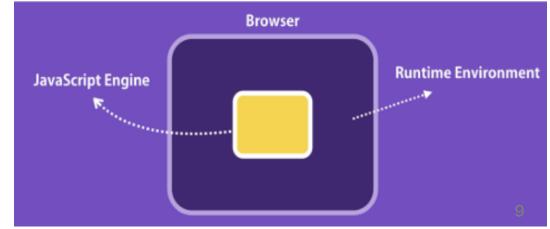
2. 자주 하는 질문

가. 자바스크립트 언어는 가장 인기있는 언어이다 https://survey.stackoverflow.co/2023/#technology

- 나. 자바스크립트 문법의 대부분은,
- 다. 자바스크립트는 어디서 실행되는가
- 라. 자바스크립트는 ECMAScript 와 뭐가 다른가? ECMAScript 는 Specification (설명서) JavaScript 는 programming language









3. 랜더링 과 파싱

가. 랜더링 (Rendering)

랜더링은 html, css, 자바스크립트로 작성된 문서를 해석 (파싱 , 문장을 문법적으로 분석하는 것) 해서 시각적 으로 출력하는 것을 말한다. (때로는 서버에서 데이터를 html로 변환해서 브라우저에게 전달하는 과정 (SSR: Server Side Rendering)를 가리키기도 한다.

나. 파싱 (parsing)

파싱은 해석하는 것이다 . 파싱은 프로그래밍 언어의 문법에 맞게 작성된 텍스트 문서를 읽어 실행하기 위해 텍 스트문서의 문자열 토큰으로 분해 (어휘분석) 하고 토큰에 문법적 의미와 구조를 반영하여 트리구조의 자료구조인 파스트리를 생성하는 일련의 과정을 말한다



4. 변수(variable)

- 가. 식별자 (identifier) : 어떤 값을 구별해서 식별할 수 있는 고유한 이름
- 나. 네이밍 규칙
 - 1) 일반적인 규칙
 - 2) 표기법
 - camelCase : 보통의 경우 이렇게 사용하면 된다 . 두번째단어의 첫자를 대문자로
 - snake_case : 언더스코어로 단어를이어준다
 - dash-case (kebab-case) : 하이픈으로 단어를 이어준다
- 다. 변수선언
- 라. 상수 (constants)
- ◆ 세미콜론 여부 { }로 만든 코드블록에는 세미콜론 X, 구문(statement)의 끝에는 세미콜론 O



5. 자료형(data type)

가. Primitive (Value, 원시타입) type

- 1) Number
- 2) String (문자열, 스트링)
- 3) Boolean (불린, 불리언)
- 4) null (널)
- 5) undefined (언디파인드)
- 6) 심볼(symbol)
- 7) BigInt

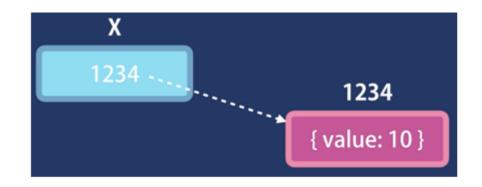
나. Reference type (객체타입)

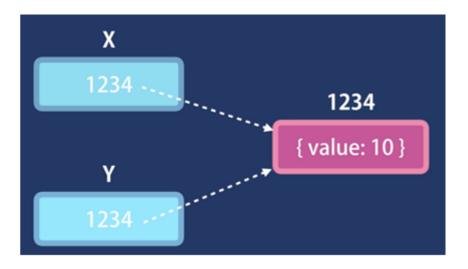
- 1)Array
- 2)Function
- 3)Object





📏 Value vs Reference Type 비교







6. 형변환

타입의 변환, 암묵적, 명시적으로 데이타타입을 변환



7. JavaScript는 무슨 일을 하는가



귀여운 – 형용사

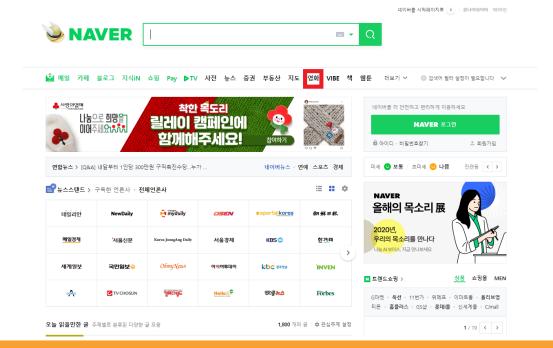
미니언이 - 명사

춤춘다 – 동사









웹 페이지에서 복잡한 기능을 구현할 수 있도록 하는 스크립팅 언어 또는 프로그래밍 언어

동적기능 동적처리 이벤트 처리 슬라이드 메뉴

2024-02-27



HTML-CSS-JS를 건축에 비유하면



[출저] https://www.deconetwork.com/blog/skinning-your-deconetwork-website/

2024-02-27



8. 자바스크립트의 위치

- html 문서를 파싱할때 from Top to Bottom 으로 진행된다
- html 문서안에서 <head> <body> 안에
- 1개의 html 파일안에 여러 개 존재할 수 있다.
- html 파일안에 또는 외부파일로 존재할 수 있다.

가. Separation of Concerns

Html (all about content) 와 js (all about behavior) 를 분리하는 것이 유지보수에 유리하다.

나. Js을 위한 <link> 속성 살펴보기

다. Script 태그안에 async / defer 속성



9. Operators 종류

- Arithmetic(산술연산자) : +, -, *, /, %, **, ++, —
- Assignment (대입): = , += , -= , *= , /= , %=
- Comparison (비교): > , < , >= , <= , ===, !== , ==
- 삼항연산자
- 논리연산자
- 쇼트서킷 개념
- Falsy 경우들



10. 조건문

가. if~ else, if ~ else if ~ else

나. switch (variable)~ case: break; ~ default:



11. 반복문 (loops)

- for
- while,
- for ~ in
- for ~ of



♥ iterable(이터러블)이란



12. break, continue

- Break는 반복문을 멈춘다
- Continue는 특정단게를 건너뛴다



13. 함수

함수는 특정작업을 수행하기 위해 독립적으로 설계된 코드집합이면서 반복적 인 내용을 수행해야 하는 경우 함수로 표현하곤 한다

- 가. 함수선언과 함수표현식
- 나. 한수의 여러가지 형태소개
- 다. 화살표 연산자의 사용 예
- 라. 함수의 arguments, length
- 마. return문



14. 내장 함수

가. 문자열 관련 메소드:

- typeof()
- includes(), startsWith(), endsWith(), indexOf(), replace(), toUpperCase(), toLowerCase(), trim(), trimLeft(), trimRight(), split('공백'), split(")

나. escape notation (/n, /')



24

15. 배열

- 배열의 생성
- 배열안의 값 변경
- Push(), unshift(), splice()
- 배열에서 엘리먼트 찾기
- 배열안에서 원하는 값이 없다는 것을 어떻게 알까?
- 배열에서 엘리먼트 제거
- 배열을 empty로 만드는 2가지 방법
- 배열 합치기 : concat(), slice(), 스프레드연산자
- 배열요소 접근하기 (forEach())
- Join(), split()
- Sort, reverse()
- 오브젝트에서 sort()

- every(), some(), filter(), map()
- 체이닝
- 같은 숫자가 몇 개인지 알아내기



실습1. JavaScript 자료형 연습하기

- 각자를 소개할 수 있는 Object 형태의 변수 선언
- 지금 까지 배운 데이터를 사용하여 자신을 소개하는 console.log 출력!

```
> console.log(me)

▼ {name: 'sean', isLeader: true, job: 'developer', interests: Array(3)} :

▶ interests: (3) ['tv', 'music', 'minion']
    isLeader: true
    job: "developer"
    name: "sean"
```



실습2. 성적 구하는 프로그램 만들기

- 그런데 결과값이 좀 이상하죠?
- Prompt 로 입력 받은 값은 "문자"로 저장이 됩니다!
 - "80" + "50" = "8050"
 - "8050" / 2 \rightarrow 4025

```
let mathScore = prompt("수학 점수를 입력 하세요");
let engScore = prompt("영어 점수를 입력 하세요");
let avg = (mathScore + engScore) / 2;
console.log(avg);
```



실습3. 형변환 실습

- 변수 mathScore, engScore 를 만들어 주세요.
- mathScore = "77"; engScore = "88";
- 시험 점수 평균을 계산하여 avgScore 에 저장하고, 이를 출력하는 프로그램을 작성하세요!
- 명시적 형 변환을 사용하여 평균 점수가 정확하게 나와야 합니다!



실습4. multifly() 함수를 만들어주세요

- 조건
 - 매개변수로 두 개의 숫자를 입력받기
 - 두 인자의 곱을 '반환' 하는 함수를 정의
 - !! 출력이 아닌 반환하는 함수
 - 콘솔창에 출력하고 싶다면 아래처럼 테스트 해보기
 - console.log(multifly(3,7)); // 21
 - console.log(multifly(2,2)); // 4



실습5. square() 함수를 만들어주세요

- 조건
 - 매개변수로 하나의 숫자를 입력받기
 - 입력받은 수의 제곱을 콘솔창에 출력하는 함수 정의
 - 콘솔창에 출력하고 싶다면 아래처럼 테스트 해보기
 - square(4)// 16
 - square(11); // 121
 - square(5); // 25



실습6. if 문 실습

• if문을 이용해서 console창에 연령대별 단어 출력해보기!

- age 변수 선언
- age가 20 이상: 성인
- age가 17 이상: 고등학생
- age가 14 이상: 중학생
- age가 8 이상: 초등학생
- age가 0 이상: 유아



실습7. 합계 구하기

• 0 ~ 100의 숫자 중에서 2 또는 5의 배수 총합 구하기!

- 힌트
 - 나머지 연산자 % 를 사용
 - 5 % 3 → 2 (5를 3으로 나눈 나머지인 2의 값을 반환)

2024-02-27



실습8. 13의 배수이자 홀수를 찾아보자

• 10000 까지의 숫자 중에서 13의 배수면서 홀수인 숫자를 찾아 봅시다!

• (심화) prompt() 를 이용해서 입력받은 수까지 13의 배수면서 홀수인 숫자를 찾는 프로그램을 만들어 봅시다!



실습9. 구구단 만들기

• 콘솔 창에 다음과 같이 출력할 수 있도록 만들어보기 (2단 ~ 9단)

2 단	<u>index7.js:105</u>
2x 1 = 2	<u>index7.js:107</u>
2x 2 = 4	index7.js:107
2x 3 = 6	index7.js:107
2x 4 = 8	index7.js:107
2x 5 = 10	index7.js:107
2x 6 = 12	index7.js:107
2x 7 = 14	index7.js:107
2x 8 = 16	index7.js:107
2x 9 = 18	index7.js:107
3 단	index7.js:105
3x 1 = 3	<u>index7.js:107</u>
3x 2 = 6	<u>index7.js:107</u>
3x 3 = 9	<u>index7.js:107</u>
3x 4 = 12	<u>index7.js:107</u>
3x 5 = 15	<u>index7.js:107</u>
3x 6 = 18	<u>index7.js:107</u>
3x 7 = 21	<u>index7.js:107</u>
3x 8 = 24	<u>index7.js:107</u>
3x 9 = 27	<u>index7.js:107</u>
4 단	<u>index7.js:105</u>
4x 1 = 4	<u>index7.js:107</u>
4x 2 = 8	<u>index7.js:107</u>
4x 3 = 12	<u>index7.js:107</u>
4x 4 = 16	<u>index7.js:107</u>
4x 5 = 20	<u>index7.js:107</u>
4x 6 = 24	<u>index7.js:107</u>
4x 7 = 28	<u>index7.js:107</u>
4x 8 = 32	<u>index7.js:107</u>
4x 9 = 36	<u>index7.js:107</u>

5 단	index7.js:105
5x 1 = 5	index7.js:107
5x 2 = 10	index7.js:107
5x 3 = 15	<u>index7.js:107</u>
5x 4 = 20	<pre>index7.js:107</pre>
5x 5 = 25	<u>index7.js:107</u>
5x 6 = 30	index7.js:107
5x 7 = 35	index7.js:107
5x 8 = 40	<u>index7.js:107</u>
5x 9 = 45	<pre>index7.js:107</pre>
6 단	<u>index7.js:105</u>
6x 1 = 6	<u>index7.js:107</u>
6x 2 = 12	index7.js:107
6x 3 = 18	<u>index7.js:107</u>
6x 4 = 24	<u>index7.js:107</u>
6x 5 = 30	<u>index7.js:107</u>
6x 6 = 36	<u>index7.js:107</u>
6x 7 = 42	<u>index7.js:107</u>
6x 8 = 48	<u>index7.js:107</u>
6x 9 = 54	<u>index7.js:107</u>
7 단	<u>index7.js:105</u>
7x 1 = 7	<u>index7.js:107</u>
7x 2 = 14	<u>index7.js:107</u>
7x 3 = 21	<u>index7.js:107</u>
7x 4 = 28	<u>index7.js:107</u>
7x 5 = 35	<u>index7.js:107</u>
7x 6 = 42	<u>index7.js:107</u>
7x 7 = 49	<u>index7.js:107</u>
7x 8 = 56	<u>index7.js:107</u>
7x 9 = 63	<u>index7.js:107</u>

8 단	<u>index7.js:105</u>
8x 1 = 8	<u>index7.js:107</u>
8x 2 = 16	index7.js:107
8x 3 = 24	index7.js:107
8x 4 = 32	index7.js:107
8x 5 = 40	index7.js:107
8x 6 = 48	<u>index7.js:107</u>
8x 7 = 56	index7.js:107
8x 8 = 64	<u>index7.js:107</u>
8x 9 = 72	index7.js:107
9 단	index7.js:105
9x 1 = 9	<u>index7.js:107</u>
9x 2 = 18	index7.js:107
9x 3 = 27	<u>index7.js:107</u>
9x 4 = 36	index7.js:107
9x 5 = 45	index7.js:107
9x 6 = 54	<u>index7.js:107</u>
9x 7 = 63	<pre>index7.js:107</pre>
0 0 70	index7.js:107
9x 8 = 72	1110CX71 31107
9x 9 = 81	index7.js:107



필기문제

- 1. "true && false" 의 결과는?
- ☐ true
- ☐ false
- □ null
- ☐ undefined
- 2. "(true && false) || true" 결과는?
- ☐ ture
- ☐ false
- □ null
- ☐ undefined
- 3. **y의 값은?**

```
let x = 10;
let y = (x > 10) ? 1 : 0;
```

- 10
- □ 1
- □ 0
- 11

4. x의 값은?

let
$$x = (2 + 3) * (4 + 5);$$

- ☐ 45
- 25
- □ 20
- □ 40

5. 결과로 true를 리턴하는 것은?

- a. '1' === 1;
- b. 1 == 1
- c. 1 === 1
- d. b and c



6. x의 값은?

```
let x = (1 == true);
```

- □ 1
- ☐ true
- ☐ false
- ☐ undefined

7. **y의 값은?**

```
let x = 10;
let y = (x > 5) && (x < 15);
```

- □ 10
- □ 5
- □ 15
- ☐ true

8. **x의 값은?**

```
let x = 5;
x += 3;
```

- □ 3
- □ 8
- □ 15
- □ 5

9. **y의 값은?**

```
let x = 10;
let y = x++;
```

- 10
- □ 11
- 12
- □ 13

10. **y의 값은 ?**

```
let x = 1;
let y = x !== 2;
```

- 1
- □ 2
- ☐ false
- ☐ ture





1번	false	4번	45	7번	true
2번	true	5번	b and c	8번	8
3번	0	6번	true	9번	10
				10번	true