**[Java Script 공부 정리]**

**\*\* 기본 문법 \*\***

**1. 변수 선언**

**- var**

: 함수 내부에서만 유효하다. 함수 밖에서 선언된 변수는 전역 스코프를 가진다.

같은 이름으로 여러 번 선언 가능하다. (재선언 가능)

선언이 함수나 전역 스코프의 맨 위로 끌어올려지며 초기화는 실제 코드 실행 시점에서 이뤄진다.

**- let**

: 블록(중괄호 ‘{}’) 내부에서만 유효하다.

같은 이름으로 변수를 재선언 할 수 없다.

호이스팅은 가능하지만, 초기화 이전에는 사용할 수 없다. (일시적 사각지대)

**- const**

: 블록(중괄호 ‘{}’) 내부에서만 유효하다.

같은 이름으로 변수 재선언 할 수 없지만 초기화 이후에는 값 변경이 가능하다.

호이스팅은 가능하지만, 초기화 이전에는 사용할 수 없다. (일시적 사각지대)

객체나 배열을 선언한 경우, 참조(reference)가 고정되기 때문에 내부의 값은 변경할 수 있다.

폰트, 스크린샷, 그래픽, 디자인이(가) 표시된 사진

자동 생성된 설명

**2. 자료형**

**- 기본 자료형**

: 변경 불가능한 값(immutable)을 가지며, 값 자체를 저장한다.

JavaScript의 기본형 자료형에는 **Number, String, Boolean, Null, Undefined, Symbol, BigInt**가 있다.

**1. Number**

: 모든 **숫자 데이터**를 포함한다. (정수, 실수)

숫자가 아닌 값을 숫자로 연산하려 할 때 ‘**NaN**’ (Not-a-Number)을 반환한다.

0을 0으로 나누기, 숫자의 범위를 넘어서는 매우 큰 숫자를 계산, Math.pow에서 큰 지수를사용하는 경우, 무한루프에서의 증가/감소 등의 경우 Infinity와 -Infinity를 나타낸다.

* Infinity: 양의 무한대. 매우 큰 숫자를 나타내며, 어떤 숫자보다도 크다.
* -Infinity: 음의 무한대. 매우 작은 숫자를 나타내며, 어떤 숫자보다도 작다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**2. BigInt**

: 큰 정수를 표현하기 위해 사용됩니다. Number 자료형으로 표현할 수 없는 **정밀한 큰 숫자를 다룰 때 유용**하다.

맨 뒤에 ‘**n**’을 붙여서 생성한다.



**3. String**

: 문자열을 나타냅니다**. 작은 따옴표('')**, **큰 따옴표("")**, 또는 **백틱(``)**으로 감쌀 수 있다.

문자열은 변경할 수 없다. (immutable)

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**4. Boolean**

: ‘**true**’와 ‘**false**’ 두 가지 값만 가진다.

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

**5. Null**

: 의도적으로 **값이 없음**을 나타낸다.



**6. Undefined**

: 변수가 **선언**되었지만 **초기화되지 않은** 상태를 나타냅니다.

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

**7. Symbol**

: **고유**하고 **변경 불가능**한 **원시 값을 생성**한다.

주로 객체의 프로퍼티 키로 사용된다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**- 객체 자료형 (참조형 자료형(reference))**

: 참조형 자료형은 객체를 저장하며, 값이 아닌 **참조를 저장**한다. **객체(Object), 배열(Array), 함수(Function)** **등**이 포함된다.\

**1. Object**

: **키-값 쌍의 컬렉션을 저장**한다. **키**는 **문자열** 또는 ‘**Symbol**’이 될 수 있으며, 값은 어떤 자료형도 가능하다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**2. Array**

: **순서가 있는 값의 집합을 저장**한다. 배열은 **인덱스로 값에 접근**할 수 있다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**3. Function**

: **재사용 가능한 코드 블록을 정의**한다. 함수도 객체이다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**4. Date**

: **날짜**와 **시간**을 저장하고 조작할 수 있는 메서드를 제공한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**5. RegExp**

: **정규 표현식**을 정의하고, **문자열의 패턴 매칭**에 사용된다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**6. Map과 Set**

: **Map**: 키-값 쌍의 컬렉션. 키는 어떤 자료형도 될 수 있다.

**Set**: 중복되지 않는 값의 컬렉션.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**3. 자료형 변환**

**- 명시적 변환 (Explicit Conversion)**

: **Number(), String(), Boolean() 등을** 사용하여 **명시적으로 변환**한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**- 암시적 변환 (Implicit Conversion)**

: **연산자**에 의해 **자동으로 변환**이 이루어진다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**4. typeof 연산자**

: ‘typeof’ 연산자를 사용하여 **변수의 자료형을 확인**할 수 있으며, 피연산자의 자료형을 문자열로 반환한다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**5. 연산자**

**- 산술 연산자**

: 산술 연산자는 **수학적 계산**을 수행하는 데 사용된다.

* **덧셈 (‘+’)**: 두 피연산자를 더한다.
* **뺄셈 (‘-‘)**: 첫 번째 피연산자에서 두 번째 피연산자를 뺀다.
* **곱셈 (‘\*’)**: 두 피연산자를 곱한다.
* **나눗셈 (‘/’)**: 첫 번째 피연산자를 두 번째 피연산자로 나눈다.
* **나머지 (‘%’)**: 첫 번째 피연산자를 두 번째 피연산자로 나눈 나머지를 반환한다.
* **거듭제곱 (‘\*\*’)**: 첫 번째 피연산자를 두 번째 피연산자만큼 거듭제곱한다.

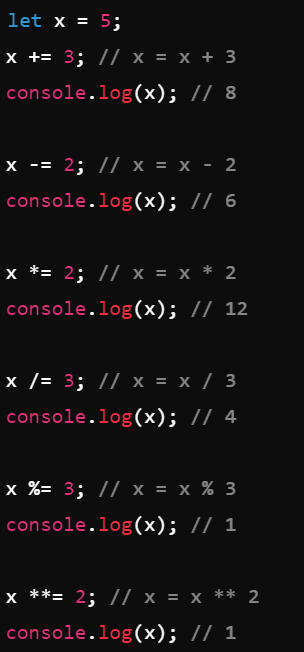
텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

- **대입 연산자 (Assignment Operators)**

: 대입 연산자는 **값을 변수에 할당**하는 데 사용된다.

* **기본 대입 (=)**: 변수에 값을 할당한다.
* **덧셈 대입 (+=)**: 변수에 값을 더한 후 다시 변수에 할당한다.
* **뺄셈 대입 (-=)**: 변수에서 값을 뺀 후 다시 변수에 할당한다.
* **곱셈 대입 (\*=)**: 변수에 값을 곱한 후 다시 변수에 할당한다.
* **나눗셈 대입 (/=)**: 변수를 값으로 나눈 후 다시 변수에 할당한다.
* **나머지 대입 (%=)**: 변수에 나머지를 구한 후 다시 변수에 할당한다.
* **거듭제곱 대입 (\*\*=)**: 변수에 거듭제곱한 값을 다시 변수에 할당한다.



**- 비교 연산자 (Comparison Operators)**

: 비교 연산자는 **두 피연산자를 비교하고 그 결과를 논리값(true 또는 false)으로 반환**합니다.

* **동등 (==)**: 값이 같으면 true를 반환합니다. (타입 강제 변환이 발생)
* **일치 (===)**: 값과 타입이 모두 같으면 true를 반환합니다.
* **부등 (!=)**: 값이 다르면 true를 반환합니다. (타입 강제 변환이 발생)
* **불일치 (!==)**: 값과 타입이 모두 다르면 true를 반환합니다.
* **크다 (>)**: 첫 번째 피연산자가 두 번째 피연산자보다 크면 true를 반환합니다.
* **작다 (<)**: 첫 번째 피연산자가 두 번째 피연산자보다 작으면 true를 반환합니다.
* **크거나 같다 (>=)**: 첫 번째 피연산자가 두 번째 피연산자보다 크거나 같으면 true를 반환합니다.
* **작거나 같다 (<=)**: 첫 번째 피연산자가 두 번째 피연산자보다 작거나 같으면 true를 반환합니다.

**텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명**

**- 논리 연산자 (Logical Operators)**

: 논리 연산자는 **논리값을 연산**하는 데 사용된다.

* **AND (&&)**: 두 피연산자가 모두 true일 때 true를 반환한다.
* **OR (||)**: 두 피연산자 중 하나라도 true이면 true를 반환한다.
* **NOT (!)**: 피연산자가 true이면 false를, false이면 true를 반환한다.

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

**- 삼항 연산자 (Ternary Operator)**

: 삼항 연산자는 **조건에 따라 값을 선택**하는 연산자이다. 조건이 **true이면 첫 번째 값을 반환**하고, **false이면 두 번째 값을 반환**한다.

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

**- 비트 연산자 (Bitwise Operators)**

: 비트 연산자는 **숫자의 비트 수준에서 연산을 수행**한다.

* **AND (&)**: 두 비트가 모두 1이면 1을 반환한다.
* **OR (|)**: 두 비트 중 하나라도 1이면 1을 반환한다.
* **XOR (^)**: 두 비트가 다르면 1을 반환한다.
* **NOT (~)**: 비트를 반전시킨다.
* **왼쪽 시프트 (<<)**: 비트를 왼쪽으로 시프트한다.
* **오른쪽 시프트 (>>)**: 비트를 오른쪽으로 시프트한다.
* **부호 없는 오른쪽 시프트 (>>>)**: 부호 없는 비트를 오른쪽으로 시프트한다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**- 문자열 연산자 (String Operators)**

: 문자열 연산자는 주로 **문자열을 결합하는 데 사용**된다.

* **문자열 결합 (+)**: 두 문자열을 결합한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**- 쉼표 연산자 (Comma Operator)**

: 쉼표 연산자는 **두 표현식을 평가**하고 **마지막 표현식의 값을 반환**한다.

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

**- instanceof 연산자 (instanceof Operator)**

: ‘instanceof’ 연산자는 **객체가 특정 클래스 또는 생성자 함수의 인스턴스인지 확인**한다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

**- in 연산자 (in Operator)**

: ‘in’ 연산자는 **객체에 특정 속성이 있는지 확인**한다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

**6. 제어문**

**- 조건문 (Conditional Statements)**

: 조건문은 **주어진 조건에 따라 다른 코드를 실행**할 수 있게한다.

* **If문**: 조건이 참일 때 코드를 실행한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* **If…else 문**: 조건이 참일 때와 거짓일 때 각각 다른 코드를 실행한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* **if...else if...else 문**: 여러 조건을 순차적으로 검사하여 첫 번째로 참인 조건의 코드를 실행한다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

* **switch 문**: 표현식의 값을 여러 경우와 비교하여 일치하는 경우에 해당하는 코드를 실행한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**7. 반복문 (Loops)**

: 반복문은 **조건이 참인 동안 코드를 반복 실행**한다.

* **for 문**: 고정된 반복 횟수만큼 코드를 실행할 때 사용한다. 초기화, 조건, 증감 부분으로 구성된다.

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

* **while 문**: 조건이 참인 동안 코드를 반복 실행한다. 반복 횟수가 정해져 있지 않은 경우에 주로 사용한다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

* **do...while 문**: 코드 블록을 먼저 실행하고, 조건을 평가하여 조건이 참이면 반복한다. 최소 한 번은 실행된다.

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

* **for...in 문**: 객체의 열거 가능한 속성을 순회할 때 사용한다. 배열에서도 사용할 수 있지만, 일반적으로 객체에 사용된다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* **for...of 문**: 배열이나 다른 이터러블(iterable) 객체를 순회할 때 사용한다. 배열의 각 요소를 반환한다.

텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

**8. break와 continue**

: 반복문 내에서 **break**와 **continue**를 사용하여 **반복문을 제어**할 수 있다.

* **break 문**: 반복문을 즉시 종료한다.

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

* **continue 문**: 현재 반복을 종료하고 다음 반복을 시작한다.

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

**9. 중첩 반복문**

: **반복문 내에 또 다른 반복문을 포함**시킬 수 있다. 주로 **다차원 배열을 처리**할 때 사용한다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**10. 레이블 문**

: 레이블 문은 **특정 코드 블록에 레이블을 붙이고 break나 continue 문에서 참조**할 수 있게 한다. 주로 **중첩된 반복문**에서 사용된다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**11. 함수**

JavaScript에서는 여러가지 방법으로 함수를 정의할 수 있다.

**- 함수 선언문 (Function Declaration)**

: 함수 선언문은 **function 키워드를 사용하여 함수를 정의**한다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**- 함수 표현식 (Function Expression)**

: 함수 표현식은 **변수에 익명 함수를 할당하여 함수를 정의**한다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**- 화살표 함수 (Arrow Function)**

: 화살표 함수는 **더 간결한 문법으로 함수를 정의할 수 있는 방법**이다. 특히 짧은 함수에 유용하다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**- 즉시 실행 함수 표현식 (Immediately Invoked Function Expression, IIFE)**

: 즉시 실행 함수는 **정의와 동시에 실행되는 함수**이다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**12. 매개변수와 인수**

: 함수는 **매개변수를 받을 수 있으며, 호출 시 인수를 전달**할 수 있다.

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

**- 기본 매개변수 (Default Parameters)**

: **매개변수의 기본값을 설정**할 수 있다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**- 나머지 매개변수 (Rest Parameters)**

: 나머지 매개변수는 **함수에 전달된 인수들을 배열로 묶을 수 있다**.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**13. 함수의 반환 값**

: 함수는 ‘**return’ 키워드**를 사용하여 **값을 반환**할 수 있다. **반환값이 없으면 undefined가 반환**된다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**14. 함수 스코프 (Scope)**

: 함수 스코프는 **함수 내에서 선언된 변수들이 함수 외부에서 접근할 수 없음을 의미**한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**15. 클로저 (Closure)**

: 클로저는 **함수가 선언될 때의 스코프를 기억하는 기능**이다. 함수가 외부 함수의 변수에 접근할 수 있게 한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**16. 콜백 함수 (Callback Function)**

: 콜백 함수는 **다른 함수의 인수로 전달되는 함수**이다. 비동기 작업에서 자주 사용된다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

**17. 함수의 다양한 형태**

**- 익명 함수 (Anonymous Function)**

: 익명 함수는 **이름이 없는 함수**로, 주로 **함수 표현식이나 콜백으로 사용**된다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**- 고차 함수 (Higher-Order Function)**

: 고차 함수는 **함수를 인수로 받거나 함수를 반환하는 함수**이다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**18. 함수의 this**

: **함수 내에서 this 키워드는 호출 맥락(context)을 참조**한다. this가 가리키는 값은 함수 호출 방식에 따라 달라진다

- **일반 함수의 this**: 일반 함수에서는 this가 전역 객체(브라우저에서는 window, Node.js에서는 global)를 가리킨다. 엄격 모드에서는 undefined가 된다.

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

- **메서드의 this**: 객체의 메서드에서 this는 해당 객체를 가리킨다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- **화살표 함수의 this**: 화살표 함수는 this를 바인딩하지 않고, 자신을 감싼 정적 스코프의 this를 참조한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**19. 함수 바인딩 (Function Binding)**

: **‘bind’ 메서드**를 사용하면 **특정 this 값을 가지는 새 함수를 생성**할 수 있다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명