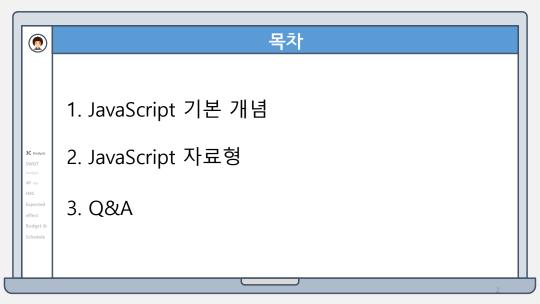
F.N.S _ 1. JS 기본 개념 2. JS 핵심 개념- 자료형

정보보호학과 20학번 전유병





● JavaScript란?

JavaScript는 동적으로 변경되는 콘텐츠를 만들고, 멀티미디어를 제어하고, 이미지에 애니메이션을 적용하는 등 거의 모든 작업을 수행할 수 있는 스크립팅 언어이다. 즉 '웹 페이지에 생동감을 불어넣기 위해' 만들어진 프로그래밍 언어이다.

3C Analysis SWOT Analysis 4P Mix IMC

● 꼭 지켜야 하는 룰

- (1) JavaScript 식별자는 '문자(a-z, A-Z)', '밑줄(_)' 혹은 '달러 기호(\$)'로 시작해야 한다. 두 번째 글자부터는 '숫자(0-9)'도 가능하다.
- (2) '대문자'와 '소문자'는 구별. 'myname'과 'myName'은 다른 이름이다.
- (3) '예약어(JavaScript가 찜해놓은 단어)'는 사용하면 안 된다. 예를 들어서 'if', 'for', 'let' 같은 것들이 있다.





console.log();

브라우저의 콘솔에 값을 출력하는 코드.

세미 콜론 (;)

- 문장을 구분하는 방법
- 다음 줄로 넘어가지 않고 한 줄에 입력할때는 ; 세미콜론으로 문장을 구분해준다.
- 문장 끝에 세미콜론을 붙이는 습관을 들여놓는게 좋다.

코멘트(주석)

- 해당 줄에 // 두 개를 붙이게 되면 주석 처리 된다.
- 줄 바꿈을 하면 풀리기에 줄 바꿈을 할 시에는 /* 코멘트내용 */ 감싸주면 된다.
- 어떤 의도로 코드가 작성되었는지 설명할 때 주로 사용을 한다.
- 구현한 코드가 어떤 동작을 하는지 설명할 때 주로 사용을 한다.

3C Analysis SWOT Analysis

4P Mix IMC Expected

effect Budget & Echadula



● 자료형 개요

숫자형 -정수, 소수

- 정수(양의 정수, 0, 음의 정수)(integer)
- 소수(float) - 사칙 연산 가능.

문자열(string)

- "Hello" →"," 따옴표로 감싼다.
- "Hello" + "yubyeong" → "Helloyubyeong" 더하기 기호로 문자열 덧셈도 가능.

불린(boolean)

- true & false 참과 거짓의 2개의 값만 가진다.

- 어떤 조건에 의한 결과 값으로 사용.
- ex) 8>6 →true, 8<6 →false

3C Analysis

4P 1/0



3C Analysis

4P 1/0

1. JavaScript 기본 개념

● 추상화 개요

- 원활한 소통을 위해 추상화를 사용.
- 구체적인 정보를 추상화 하여 꼭 필요한 핵심 만을 뽑아냄.
- 복잡한 것들을 목적에 맞게 단순화 하는 것.

변수

- 값에 이름을 부여하는 것.
- 변수 선언(let 사용) - let 변수이름= 값;
- 변수를 사용해서 코드를 추상화.

함수

함수 선언

function 함수이름() {

명령; 명령;

변수 = 값, 함수 = 명령들 콘솔에 값을 출력하기 위해 사용하는console.log 또한 함수 함수 호출

함수이름();



● 파라미터

- 함수를 호출할 때 소괄호 안에 입력하는 값을 파라미터 (매개변수)라고 한다.
- 함수 선언에서 함수이름 뒤에 오는 소괄호에 파라미터를 작성하고 함수 내부에서 파라미터를 변수처럼 사용하면 함수를 호출할 때 소괄호에 입력한 값이 파라미터에 전달된다.

함수 선언

- function 함수이름(파라미터 →값) {
 console.log(파라미터 → 값);
}

" 함수 호출

- 함수이름(값);
- 호출할 때 소괄호에 입력한 값이 함수의 파라미터로 전달된다.

여러 개의 파라미터 사용 가능

- function 함수이름(파라미터1, 파라미터2) {};

함수이름(파라미터1, 파라미터2); →호출

return문

- 함수의 output의 해당되며 함수 내부에서 정해 진 명령을 실행하고 값을 반환한다는 뜻이다.
- 함수 내부에서 사용
- return 리턴값;

3C Analysis SWOT

Analysis

4P Mix

IMC

Expected

effect Budget & Schedule



숫자형

- 나머지 연산 : %
- 거든제곱·**
- is도 연산의 우선 순위 따름. (괄호 안의 연산 우선, 덧셈보다 곱셈이 먼저 계산,etc)

문자열

3C Analysis 4P 1/0

- 따옴표(', ") 로 감싸줘야 함.(출력 시 따옴표가 같이 출력되지는 않음)
- 따옴표를 기호로 쓰고 싶을 때는 따옴표 앞에 역 슬래시₩ 사용. - `도 사용 가능. 문자열 전체를 `로 감싸준다.(역 따옴표)
- ex) 'He said "I'm yubyeong" -> He said "I'm yubyeong"
- 문자열 연산 가능
 - console.log("hello" + "world") → helloworld
 - console.log("3" + "5") → 35



불 대수

- 일상적인 논리를 수학적으로 표현한 것.
- 불 대수의 값은 True & False 두 값만 존재.
- 불 대수의 연산은 and or not 세가지 연산 존재.
- and 연산
- x와 y가 모두 참일 때만 x and y 가 참
- or 연산
 - x와 y가 하나라도 참이면 x or y는 참
- not 연산
 - not 은 반대로 돌리는 것이다.
 - ex) not 2는 1보다 작다. → 2는 1보다 작지 않다. 라는 뜻이므로 True이다.

불린형

- 불린형의 값은 true와 false 명제에 따른 값을 출력한다.
- **크거나 같다** → >= (등호를 항상 부등호 뒤에 작성.) **작거나 같다** → <=
- 값이 일치한다 → === 불일치 → !==
- and 연산 \rightarrow && or 연산 \rightarrow \parallel not 연산 \rightarrow ! (ex. !true \rightarrow false, !false \rightarrow true, !!true \rightarrow true)

3C Analysis

Analysis 4P Mix IMC

> effect Budget &



typeof 연산자

- typeof 값 → 이 값의 자료열을 문자열로 되돌려줌.
 - ex) console.log(typeof 101); → number
 - console.log(typeof 'yubyeong'); → string
 - console.log(typeof true); → boolean
- 소수와 정수를 구별하지 않기에 숫자형은 모두 number로 되돌려줌.
- 함수 또한 function으로 되돌려줌.
- 연산을 할 때는 괄호를 씌어주면 됨.(연산의 우선순위를 올려준다.)
 - console.log(typeof (5 + 3)); → number
 - 잘못된 예시) console.log(typeof 5 + 5)); → NaN(not a number)

4P My

3C Analysis



형 변환 (Type Conversion)

- 변환하고 싶은 형으로 함수를 사용
 - String, Number, Boolean
- ex) console.log(Number("10") + Number("5")); → 15
- 불린을 숫자형으로 형 변환하면 0과 1로 값이 나옴.
- true → 1, false →0
- 문자와 숫자형을 불린형으로 형 변환하면 기본적으로 true값을 되돌려줌.
- false 값을 되돌려주는 경우
 - 문자형은 따옴표만 있는 빈 문자열
 - 숫자형은 0 , NaN
- 산술 연산은 숫자로 자동으로 변환하여 계산.
 - ex) console.log(4 true); \rightarrow 3
- 같음 비교 연산자에서 ===은 일치 , ==은 동등.
 - -동등 연산자(==)는 두 피연산자의 타입이 서로 다르면, 비교를 위해 강제로 타입을 같게 변환.
 - -일치 연산자(===)는 타입의 변환 없이 두 피연산자의 값이 같고, 타입도 같아야만 참(true)을 반환.

- But 문자열 더하기는 문자열이 더 우선이다. console.log(4 + '2'); → 42 문자열로 계산이 되는데 이렇듯 자바스크립트에서 더하기 기호는 숫자를 더하는 것보다 문자열을 연결하는 기능이 더 강하다고 생각하면 된다. 더하기를 할 때 순서에 상관없이 어느 한쪽이라도 문자열이 있다면 양쪽 모두를 문자열로 바꾼 다음 문자열 연산으로 동작을 한다.

3C Analysis SWOT Analysis

Analysis

4P Mix

IMC

Expected

effect Budget & Schedule



템플릿 문자열 (template : 일정한 틀, 형식)

- ` 을 사용하면 된다.
 - ex) console.log(`생년월일은 \${year}년 \${month}월 \${day}일 입니다.`);
- → 생년월일은 2023년 11월 06일 입니다. (여기서 \${}안에 있는 year month day는 본인이 직접 지정 한 변수값)

null과 undefined

- 자료형에 해당한다. → '값이 없다'를 나타냄.
- 선언을 하고 아무런 값을 지정 하지 않으면 undefined.
- **의도적으로** 값이 없다고 지정을 해줄 때는 null.
- undefined는 값이 지정되지 않은 경우를 의미하지만, null은 해당 변수가 어떤 객체도 가리키지 않는다를 의미. ※ typeof를 통해 자료형을 확인해보면

※ typeof을 중에 자료영을 확인해보면 null은 object로, undefined는 undefined가 출력. let year = 2023; let month = 11; let day = 06;

console.log ("생년원일은 " + year + "년 " + month + "월 " + day + "일 " + "입니다.")

-문자열을 나눠서 더하기 기호(+)를 통해 연 결할 필요가 없기때문에 훨씬 가독성이 좋다.

Analysis

4P Mix

IMC

Expected

3C Analysis

Expected effect Budget &



감사합니다.