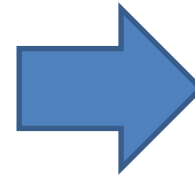


객체(Object)

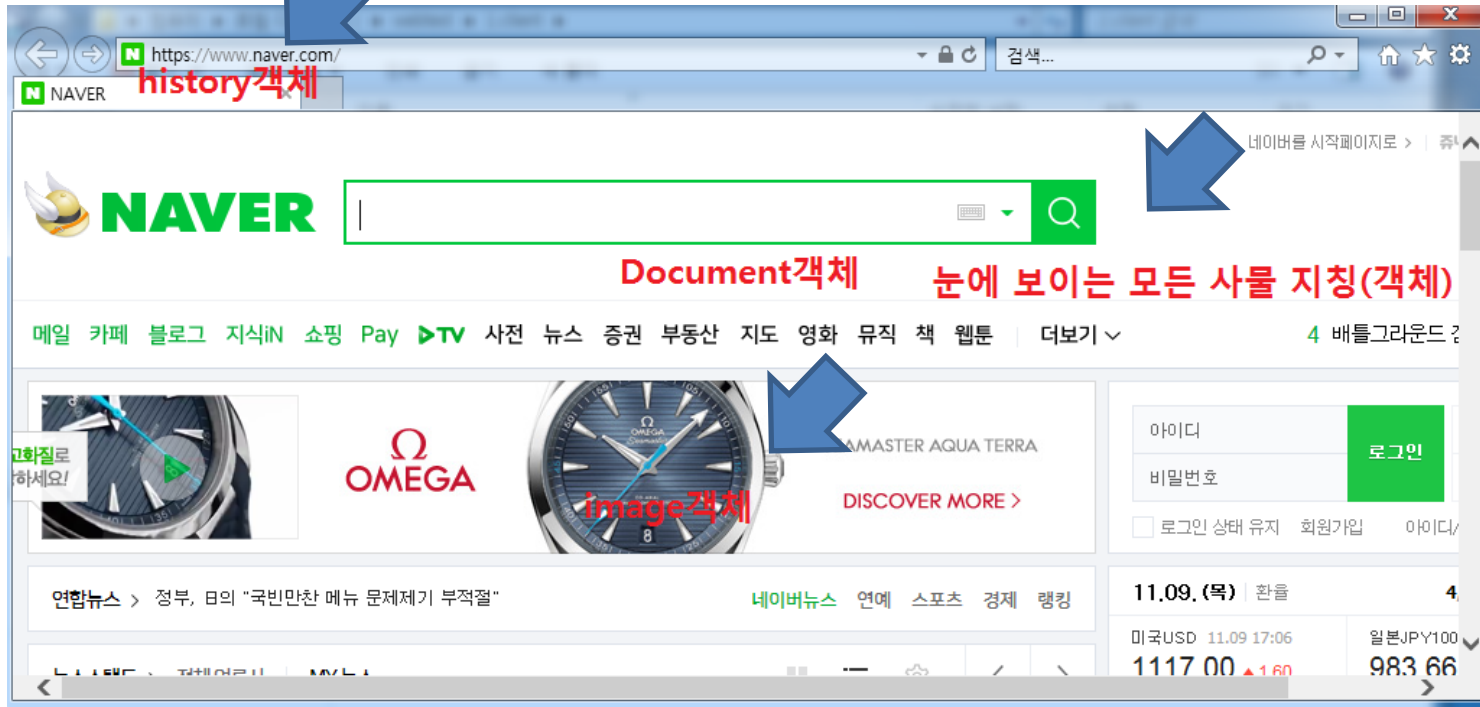
객체 => 눈에 보이는 모든
사물을 지칭하는 말



객체생성

=> 데이터를 저장할 공간을 만드는 것
(ex 자동차, 집, 컴퓨터,,,모든 사물)

자바스크립트의 객체



Window객체

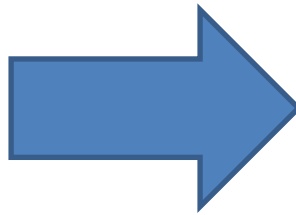
객체의 구성요소(집)

- 1.속성
- => **데이터 저장목적**
- => **객체명.속성=저장할값**
- 2.함수
- => 멤버변수를 통해
- **저장된 데이터를 조회,수정,삭제하는**
- **기능(TV 시청->리모콘이용)**



함수의 호출 유형

지시 내림(매개변수를 전달)



작성한 내용을 보고
(호출한 쪽으로 되돌려줌
=> 반환값 존재)

Caller Function

(특정 동작을 수행하는 함수를
호출하는 함수
직장상사)

Worker Function

(caller Function에 의해서
호출되어서 실제로
특정작업을 수행하는
함수)=> 보고서 작성
(실행결과를 알려줌)
-> 리턴값, 반환값

함수의 개요

1. 어떤 기능을 가진 최소한의 작업단위
(계산목적, 데이터 저장목적, 데이터조회,
출력목적...)

2. 함수의 구분

- 1) 사용자정의 함수->객체가 생성=>호출가능
객체 없이 단독 호출가능
- 2) 내장함수->객체명.함수명()->객체명 생략

함수의 구성방식(반환값유무)

- function 함수명(변수, 변수2,...){
- 함수의 주요기능 작성(변수, 제어문)
- }
-
- function 함수명(변수, 변수2,...){
- 함수의 주요기능 작성(변수, 제어문)
- return 반환값(수식, 변수,,)
- }

함수의 적용예1

(매개변수, 반환형이 없는 경우)

```
function printTest(){
```

함수명()

```
    document.write("테스트1")
```

```
    document.write("테스트1")
```

```
    ....
```

```
}
```

```
function printTest(){
```

```
    for(let i=1;i<=10;i++)
```

```
        document.write("테스트"+i)
```

```
}
```

자주 사용이 되고, 반복적인 처리를 해야하는 코드가 필요한 경우

가능한 적은양의 소스코드를 사용할 수록 좋다

함수의 적용예 2

(매개변수가 있고, 반환값이 없는 경우)

객체명.printTest(100);

function printTest(**n**) {
 for(let i=1;i<=n;i++)
 document.write("테스트"+n)

외부로부터 값을 입력
받아서 값을 계산, 출력, 저장 목적
으로 사용한다.

}

객체명.printTest(1,10);

function printTest(**start,end**) {
 for(let i=start;i<=end;i++)
 document.write("테스트"+i)

}

테스트5
테스트6
테스트7
테스트8
테스트9
테스트10

함수의 적용예 2

(매개변수가 있고, 반환값이 있는 경우)

입력받을값(id값)
ex) String id

입력받을값(password값)
ex) String password

아이디 또는 비밀번호 오류입니다.
현재 입력하신 아이디가 네이버에 등록되어 있지 않거나, 아이디 또는 비밀번호를 잘못 입력 하셨습니다.

1 2 3 보안 2 단계 IP보안 ON

aaa

비밀번호

로그인

☐ 로그인 상태 유지

로그인

nup49rok1

로그인

☐ 로그인 상태 유지

function login(id,pw)

내부적으로 id,password
존재하는지 확인절차

nup49rok1 님

로그아웃

메일 31 | 쪽지 33 | 마이리지 소개보기

me 홈 알림 1 메일함 31 구독함 바로가기

false (id ,password가
존재하지 않는 경우)

계산결과값
(리턴값)
boolean

true (id ,password가
존재하는 경우)

메서드내부의 처리결과
를 화면에 보여줄수 있
도록 해주는 값

함수의 적용예 3

(매개변수, 반환값 필요한 경우)

