자바스크립트 개념 정리

**1. 변수**

What is : 변수는 데이터들을 담을 수 있는 컨테이너

When to use : 변할 수 있는 값을 담을 때 사용.

**2. 할당**

What is : 변수에 값을 할당하는 행위

When to use : 변수에 값을 할당해주거나 변수에 할당된 값을 수정하고 싶을 때 사용

**3. 함수**

What is : 실제 동작하는 코드 블록이다.

When to use : 반복해서 사용될 동작에 대해 정의할 때 사용되며 다른 인자 값을 배정하여 다른 결과값을 받고 싶을 때 사용.

**4. 리턴**

What is : 함수가 호출되었을 때 호출되어진 곳으로 특정 값을 반환

When to use : 함수가 호출되어 읽혀지는 중 실행을 중단하게 하거나,

함수가 실행되었을 때 호출된 곳으로 어떠한 값을 돌려보내고 싶을 때 사용.

**5. 객체**

What is : 이름과 값이 짝지어진 데이터를 담는 컨테이너

When to use : 이름과 값을 짝지어서 관리하면 효율적인 데이터를 관리할 때 사용.

**6. 이벤트**

What is : html 요소들에게 일어나는 동작(유저가 혹은 browser가 주체가 될 수 있다.)

When to use : html 요소들에게 특정 동작이 일어나는 시점을 포착하고 싶을 때 사용

**7. 배열**

What is : 하나의 컨테이너에 여려개의 값을 할당하고 싶을 때 사용

When to use : 굳이 할당된 값에 이름을 붙여서 관리할 필요가 없을 때 사용

**8. if 문(if 문, else if 문, else 문)**

What is : 상태에 따라 다른 행위를 발생시키고 싶을 때 사용되는 문장

When to use : 상태에 따라 다른 행위를 발생시키고 싶을 때 사용

**9. switch문**

What is : 상태에 따라 다른 행위를 발생시키고 싶을 때 사용되는 문장

When to use : 상태에 따라 발생시켜주고 싶은 행위가 다양할 때 사용

**10-1. break**

What is : 문장이 읽히는 중에 break를 만나면 문장을 읽는 행위가 중단되고 코드 블록 밖으로 나가진다

When to use : 문장에서 실행을 중단시켜 주고 싶을 때 사용

Question : return으로 실행을 중단시키는 것과 구별해서 사용되어지는 것인다?

그게 아니라면 switch문에서 break를 쓰는 것은 관습적인 것일 뿐인가?

**10-2. continue**

What is : break문은 loop를 중단 시키는 반면 continue는 특정 상태일 때 실행을 건너뛴 후 loop문은 계속 진행한다.

When to use : 중단점이 끝에 있지 않고 중간중간의 상태를 체크하며 동작을 수행해 나가야 될 경우에 사용

**11. for문**

What is : 상태에 따라 행위를 여러 번 반복해주는 문장

When to use : 행위를 여러 번 반복해주고 싶을 때 사용

**12. for/in 문**

What is : 배열 또는 객체 내부의 속성의 개수만큼 특정행위를 반복시켜줌

When to use :

- 객체 배열 모두 상용 할 수 있다.

- Key 값과 value 둘다 출력 할 수 있다.

- 배열 또는 객체 내부의 속성의 개수만큼 특정행위를 반복시켜 주고 싶을 때 사용

**13. for/of 문**

What is : 배열 내부의 속성의 개수만큼 특정행위를 반복시켜줌

When to use :

- 배열에서만 사용 되어진다.

- Index num은 반환하지 않고 배열에 할당된 값만 반환한다.

- forEach에서 콜백함수를 사용해야 하는 불편함을 없애기 위해 나왔다.

**14. while 문**

What is : 상태에 따라 행위를 여러 번 반복해주는 문장

When to use : 행위를 여러 번 반복해주고 싶을 때 사용

Question : for 문과 차이를 모르겠다 for 문이 더 쓰기 편한데 for문을 쓰면 되는건가?

**15. Do/while 문**

What is : 실행문을 무조건 1회 실행 후 상태에 따라 행위를 여러 번 반복해주는 문장

When to use : 특정 동작을 무조건 1회 실행 한 후 경우에 따라 반복시켜주고 싶을 때 사용.

Question: 실제로 사용할 일이 있을지 잘 모르겠다.

**16. errors object**

What is : 내장된 에러 객체가 있어서 에러발생시 이름과 메시지를 반환해준다

When to use : console창에 보면 나오는 에러메시지

**16-1. errors - try**

What is : 문장에 error 가 포함되어 있는지 시험 동작시킨다

When to use : 문장에 오류가 있는지 테스트 할 때 사용

**16-2. errors - catch**

What is : try 문에서 error 발생 시 catch 문에서 특정 동작을 발생 시켜준다.

When to use : 오류가 발생할 경우 어떤 문제인지 가시화 할 수 있다.

**16-3. errors - throw**

What is : try 문 안에서 error 발생시 String, a Number, a Boolean or an Object를 반환 시킬 수 있다.

When to use : 에러가 발생하는 시점을 구분하여 문제되는 부분을 구체화 시킬 수 있다.

**16-4. errors - finally**

What is : try 문과 catch 문의 결과와 상관없이 마지막으로 실행해주는 문장

When to use : 초기화에 사용될 수 있어 테스트를 간편하게 해준다.

**17. scope**

What is :

변수들의 접근성을 결정하며 전역 스코프와 지역 스코프가 있다.

자바스크립트의 모든 함수는 스코프가 있다.

When to use : 변수가 사용되어질 범위를 정해준다.

**18. 전역변수**

What is :

- 함수 밖에서 선언된 변수

- 함수 안에서 선언되지 않고 생성된 변수(var 혹은 new 없이 생성)

When to use : 어디서든 접근 가능한 값을 생성할 때 사용

**19. 지역변수**

What is : 함수 내부에서 선언된 변수

When to use : 독립적으로 사용되어 변수 내부의 값이 오염되지 않아야 될 시 사용

**20. 끌어올림(hoisting)**

What is :

- 자바시크립트가 선언을 가장 상위로 옮기는 행위

- 변수의 선언은 이와 같은 행위가 적용 되지만, 할당은 적용되지 않는다.

- let 과 const 로 선언된 변수에서는 적용되지 않는다.

-

When to use : hoisting으로 인한 혼돈을 방지하기 위해 변수 선언은 해당 scope의 최 상단에서 해야한다..

**21. 엄격모드(strict mode)**

What is :

- ecma 5 에서 등장

- ie10부터 사용가능

- 안정적인 자바스크립트 코딩이 가능(나쁜 코딩 기법들을 에러메시지로 받아들임)

What is :

- 선언 : 선언 없이 변수 생성 불가

- 선언 : eval, arguments, implements, interface, let, package, private, protected, public ,static, yield

등 예약어로 변수선언 불가

변수 선언 불가

- 할당 : 8진법 숫자 할당 불가

- 할당 : 8진법 escape 문자 할당 불가

- 수정 : readonly 프로퍼티에 값 할당 불가

- 수정 : get-only 프로퍼티에 값 할당 불가

- 삭제 : undeletable 프로퍼티(prototype) 삭제 불가

- 호출 : global scope에서 this가 포함된 함수 호출 시, 일반모드에서는 this에 window객체가 **리턴**, 엄격모드에서는 undefined가 리턴된다.

When to use : 코드를 안정적으로 작성하고 싶을 때 사용한다.

23. this

What is : 함수를 호출한 객체를 언급한다

How to use:

- method 안에서 this는 메서드를 포함하는 주인 객체다

- 혼자 쓰일 때 this는 global 객체다

- 함수에서 this는 global 객체다

- 엄격 모드 시 함수에서 this는 global 객체다

- 이벤트에서 this는 이벤트가 발생하는 객체다

- call() 혹은 apply() 에서 this는 어떠한 객체도 가능하다.

When to use : 호출자가 누구인지 알아야 될 때 사용

**To do list**

- 클로저, 형 변환, this, classes, prototype, call(), apply()

- errors object 발생시 name과 message 검색해서 내용 확인

3.

What is :

When to use :

**var** basketModule = (**function**(){

...

})()

**var** basketModule = **function**() {

...

}  
혹시 모르게 변수의 스코프가 외부로 흘러나가게 되어 전역상태의 변수의 상태를 어지럽히게 됨을 완전히 차단합니다. 위의 코드는 완벽하게 private 한 상태가 됩니다.  
클로저 –

Return 문 안에 객체를 넣는 경우와 그냥 함수 안에서 객체를 지정하는 것의 차이

**네임스페이스 패턴**

// 위험함 패턴  
**var** MYAPP = {};  
  
// 개선된 패턴  
**if**(**typeof** MYAPP === "undefined"){  
 **var** MYAPP = {};  
}  
  
//또는 더 짧게 쓸 수 있다.  
**var** MYAPP = MYAPP || {};