Use strict 모드

-더 나은 오류 검사를 코드에 사용하는 방법

Const : 상수(값이 변하지 않는 변수)

Let : 함수 안에랑 밖에 변수 이름이 같으면 잘못된 값이 나올 수 있기 때문에 이를 방지하는 변수

Var : 일반 변수

과제

-스크립트 자료형에 대해서 자료조사

-var, let, const 키워드 조사, 차이점에 대한 예제 첨부하여 메일발송

[zerocooldog@pionnet.co.kr](mailto:zerocooldog@pionnet.co.kr)

비교 연산자

==, != 값만 비교

===, !== 타입까지 비교

특수 연산자(delete)

Var를 이용해서 선언된 변수는 지울 수 없음.

객체, 개체의 속성, 배열의 특정 인덱스에 있는 원소를 지우는 연산자

특수연산자(in)

지정된 객체에 해당되는 속성이 있으면 true를 반환

사용 방법 : 속성 in 객체

특수연산자(instanceof)

-개체 이름 instanceof 개체 형식(Date, Array)

-개체가 해당 형식이면 true를 반환

특수연산자(typeof)

표현식

-Try-catch-finally

함수표현문

-함수표현문 : var로 선언

-변수에 함수를 지정, 런타임 시에 변수에 할당된다. 주의 요망

-함수 표현문의 경우 선언하기전에 호출하면 undefined

1. 익명 함수 var foo=function(){console.log(‘hello’);};
2. 기명 함수 var foo=function foo(){console.log(‘hello’);}; -> 거의 안씀
3. 즉시 실행 함수 –함수가 선언 되자마자 실행됨 (function foo(){console.log(‘hello’)})(2,3);

-scope & scope chain

-클로저(closure)

Java의 private과 같은 역할

Why? Java의 캡슐화를 사용하는 이유와 동일함

\*객체

-자바 스크립트 객체

1. 사용자 정의 개체

2. 내장 객체 (String, Date,Array, Number)

3. 브라우저 내장 객체 – 브라우저 수행 시 생성(location, document,history)

객체 value 값 읽는법

-Flight[“airline”]

-Flight.airline

Prototype : 참조의 개념

내장객체

History 객체

Location 객체

DOM

* HTML과 XML 문서에 대한 프로그래밍 인터페이스
* 문서에 대한 구조적 정보를 제공
* 웹 페이지를 스크립트나 다른 개발 언어로 접근할 때 필수적

-css 조작하기

1) 단일 속성 변화

Var div=document.getElementById(“id”);

div.style.color=”red”;

2) 여러 속성 변화

Var div=document.getElementById(“id);

div.style.cssText=”color:red; border:1px solid green”;

css

-style.classList

DOM 추가/변경/제거

Appendchild , prependchild

브라우저는 인터프리터언어(한줄한줄씩 차례대로 처리)

Onload 역시 $(document).ready()처럼 페이지를 다 읽고 나서 처리함

Selector

Element Selector - $(“elementName”) -> Tag

on() 메서드를 이용하는 이유 - 동적으로 사용하기 위해(신규 추가등의 경우)