JavaScript

실행 컨텍스트 – 김근형 강사

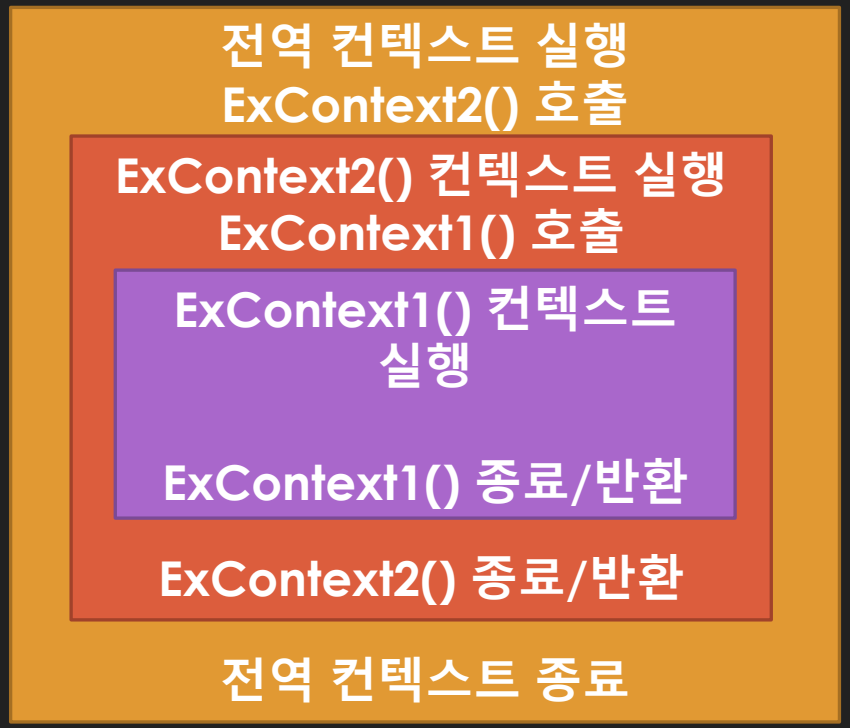
실행 컨텍스트

- 실행 컨텍스트 개념
  - 실행 가능한 코드를 형상화하고 구분하는 추상적인 개념
  - 실행 가능한 자바스크립트 코드 블록이 실행되는 환경
  - 실행 컨텍스트가 형성되는 경우를 세 가지로 규정하면 전역 코드, eval() 함수로 실행되는 코드, 함수 안의 코드를 실행할 경우가 있다.
  - 현재 실행되는 컨텍스트에서 이 컨텍스트와 관련 없는 실행코드가 실행되면, 새로운 컨텍스트가 생성되어 스택에 들어가고 제어권이 그 컨텍스트로 이동한다.

실행 컨텍스트

- 실행 컨텍스트 예제

```
console.log("This is global context");

function ExContext1() {
    console.log("This is ExContext1");
};

function ExContext2() {
    ExContext1();
    console.log("This is ExContext2");
};

ExContext2();
```



실행 컨텍스트 생성 과정

- 아래의 자바 스크립트 소스를 참고하여 실행 컨텍스트 생성 과정을 설명한다.

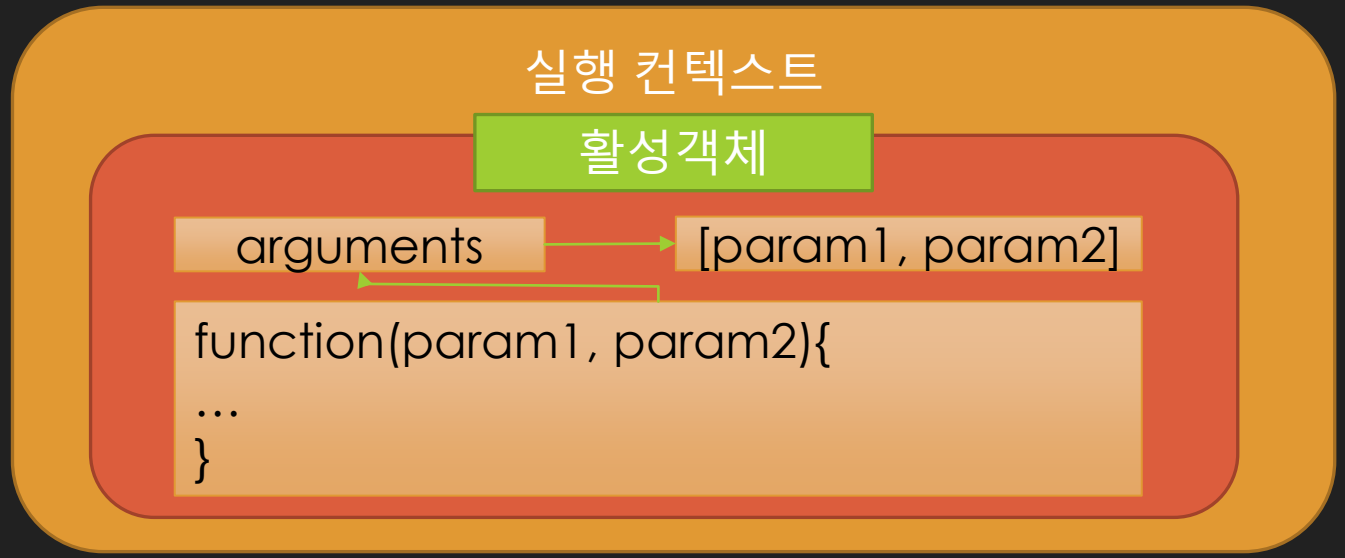
```
function execute(param1, param2) {
    var a = 1, b = 2;
    function func() {
        return a+b;
    }
    return param1+param2+func();
}

execute(3, 4);
```

실행 컨텍스트 생성 과정

- 활성 객체 생성
  - 활성 객체 : 실행에 필요한 여러 정보를 담은 객체
  - 실행 컨텍스트가 생성되면 자바스크립트는 해당 컨텍스트에서 활성 객체를 생성한다.
  - 이 객체에 앞으로 사용하게 될 매개변수나 사용자가 정의한 변수 및 객체를 저장하고, 새로 만들어진 컨텍스트로 접근 가능하게 되어있다.
  - 이는 엔진 내부에서 접근할 수 있다는 것이지 **사용자가 직접 접근할 수 있다는 것은 아니다.**

실행 컨텍스트 생성 과정

- arguments 객체 생성
  - 앞서 만들어진 활성 객체는 arguments 프로퍼티로 이 arguments 객체를 참조한다.

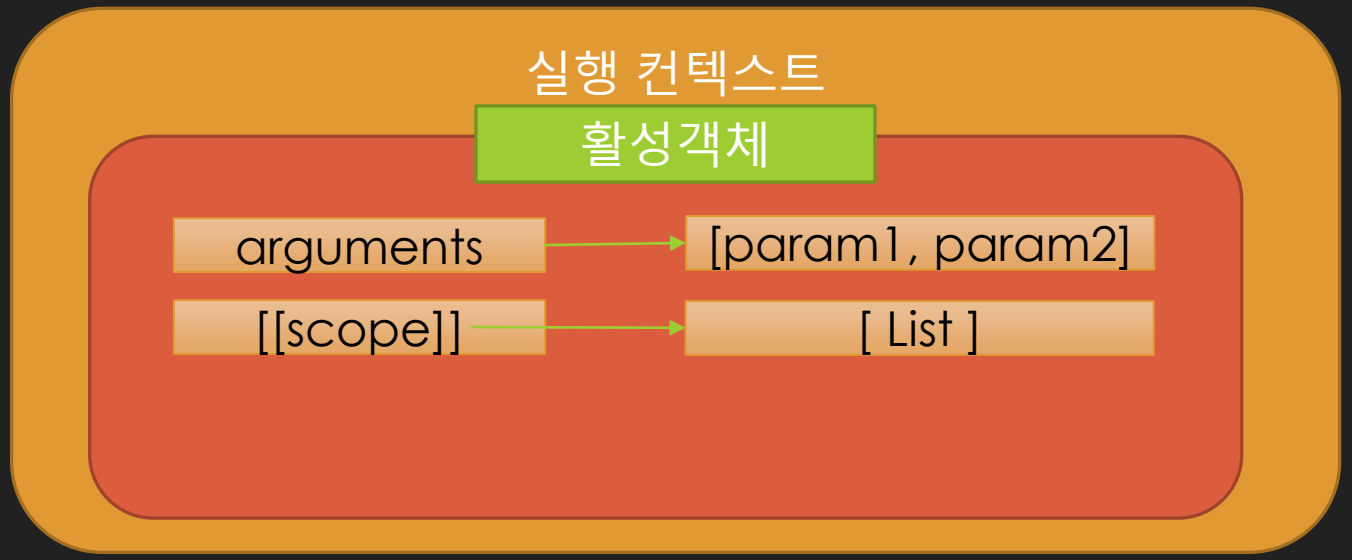


실행 컨텍스트 생성 과정

- 스코프 정보 생성
  - 현재 컨텍스트의 유효 범위를 나타내는 스코프 정보를 생성한다.
  - 이 스코프 정보는 현재 실행중인 실행 컨텍스트 안에서 연결 리스트와 유사한 형식으로 만들어진다.
  - 현재 컨텍스트에서 특정 변수에 접근해야 할 경우, 이 리스트를 활용한다.
  - 이 리스트로 현재 컨텍스트의 변수 뿐 아니라, 상위 실행 컨텍스트의 변수도 접근이 가능하다.
  - 이 리스트에서 찾지 못한 변수는 결국 정의되지 않는 변수에 접근하는 것으로 판단하여 에러를 검출한다.
  - 이 리스트를 **스코프 체인**이라 하는데 [[scope]] 프로퍼티로 참조된다.
  - 스코프 체인을 통해 내부 함수에서는 외부 함수의 변수에 접근 가능하지만 외부 함수에서는 내부 함수의 변수에 접근할 수 없다.
  - 활성 객체는 스코프 체인의 제일 앞에 추가되며, execute() 함수의 인자나 지역 변수 등에 접근할 수 있다.

실행 컨텍스트 생성 과정

- 스코프 정보 생성

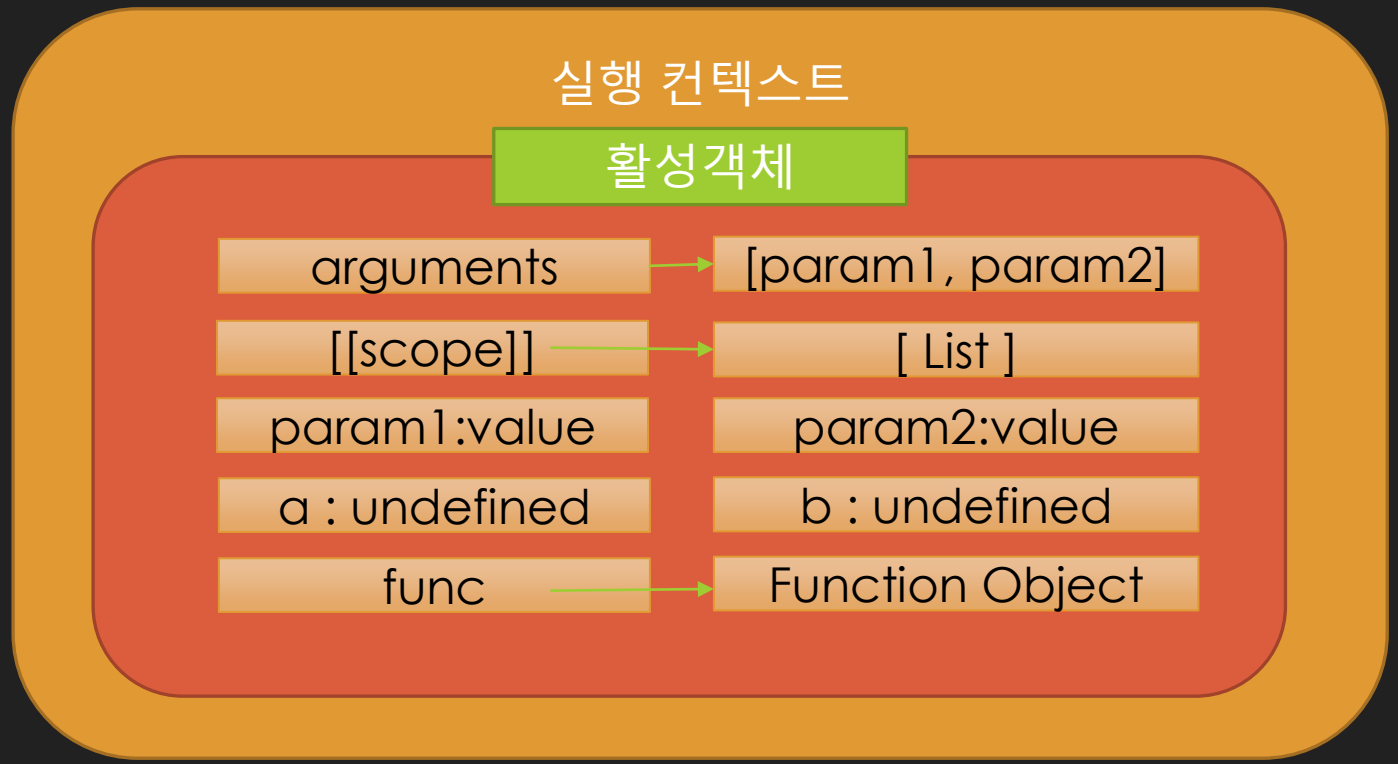


실행 컨텍스트 생성 과정

- 변수 생성
  - 현재 실행 컨텍스트 내부에서 사용되는 지역 변수의 생성이 이루어진다.
  - 앞에서 생성된 활성 객체가 변수 객체로 사용된다.
  - 변수 객체 안에서 호출된 함수 인자는 각각의 프로퍼티가 만들어지고 그 값이 할당된다.
  - 값이 넘겨지지 않을 경우 undefined가 할당된다.
  - execute() 함수 안에 정의된 변수 a, b와 func가 생성된다.
  - 주의할 점은 이 과정에서 변수나 내부 함수를 단지 메모리에 생성하고, 초기화는 각 변수나 함수에 해당하는 표현식이 실행되기 전까지 이루어지지 않는다. 그래서 a와 b는 undefined로 할당된다.

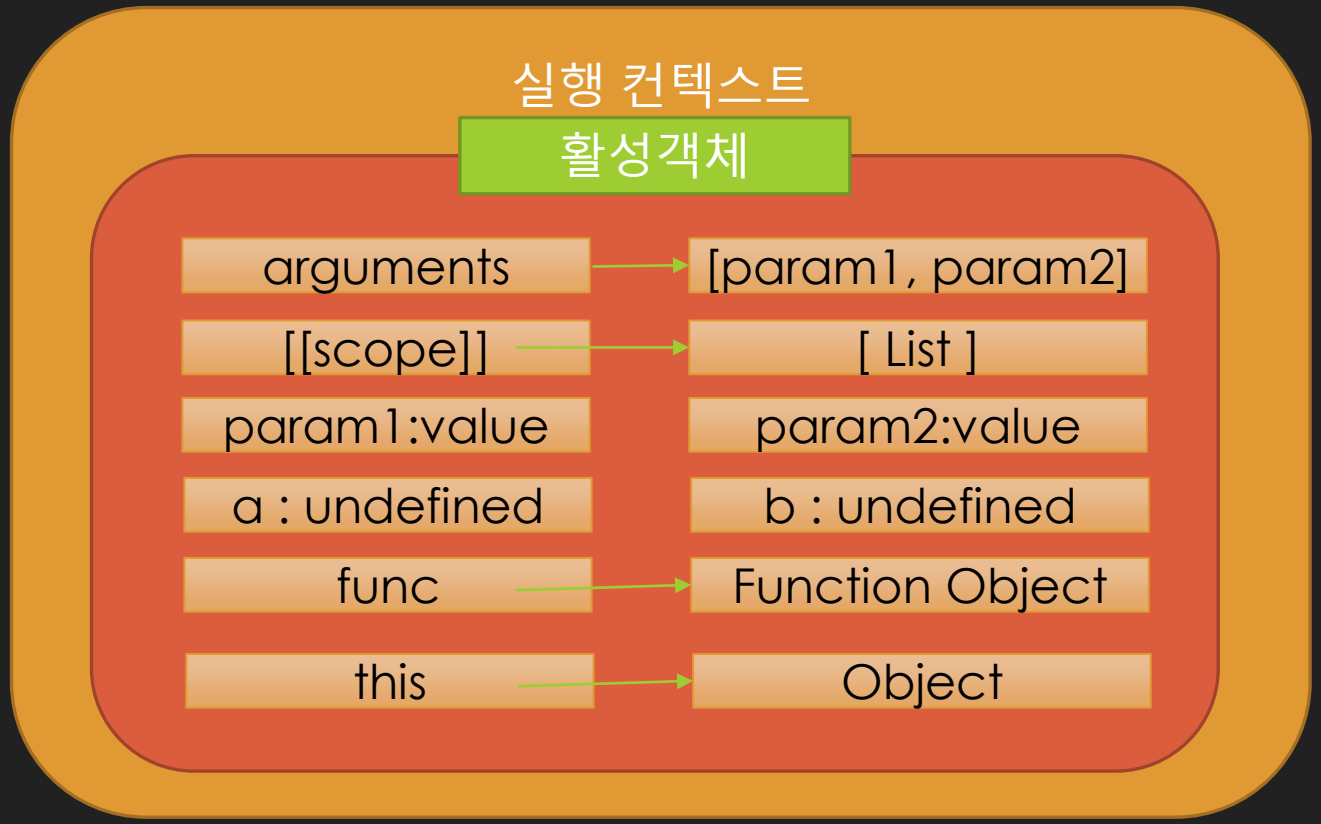
실행 컨텍스트 생성 과정

- 변수 생성



실행 컨텍스트 생성 과정

- This
  - 마지막 단계에서 this 키워드를 사용하는 값이 할당된다.
  - 여기서 this가 참조하는 객체가 없으면 전역 객체를 참조한다.



실행 컨텍스트 생성 과정

- 코드 실행
  - 하나의 실행 컨텍스트가 생성되고, 변수 객체가 만들어진 후에 코드에 있는 로직이 실행된다.
  - 이렇게 실행되면서 변수의 초기화 및 연산, 또 다른 함수 실행 등이 이루어진다.
  - 전역 컨텍스트는 arguments 객체가 없으며 전역 객체 하나만을 포함하는 스코프 체인이 있다.

함수 스코프 및 실행 컨텍스트 예제(내부 함수)

```
// 2-4-1
var a = 1;
function outer() {
  console.log(a);

  function inner() {
    console.log(a);
    var a = 3;
  }

  inner();

  console.log(a);
}
outer();
console.log(a);
```

함수 스코프 및 실행 컨텍스트 예제(메소드)

```
// 2-5-1
var obj = {
  a: 1,
  b: function bb() {
    console.log(this);
  },
  c: function() {
    console.log(this.a);
  }
};

obj.b();
obj.c();

console.dir(obj.b);
console.dir(obj.c);
```

스코프 & 실행 컨텍스트 요약

스코프 유효범위 (변수)
SCOPE

실행 컨텍스트 실행되는 코드덩어리
EXECUTION CONTEXT (추상적 개념)