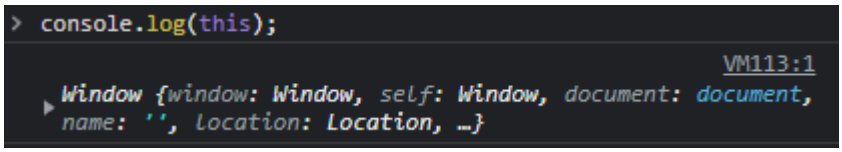
JS-this

태그	
날짜	@2023년 4월 27일

this → { ? } : 어떤 객체를 가르키는 키워드

this가 가르키는 객체는 상황에 따라 달라진다.

“this는 함수를 호출한 객체이다”



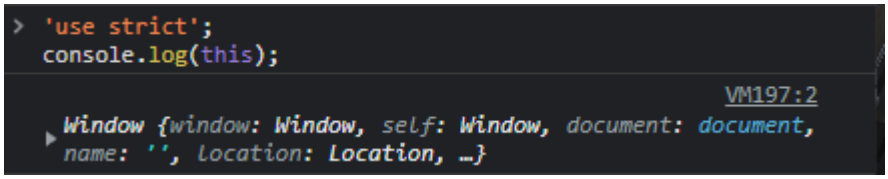
window의 객체는 브라우저의 대한 객체를 가지고 있는 전역객체이다.

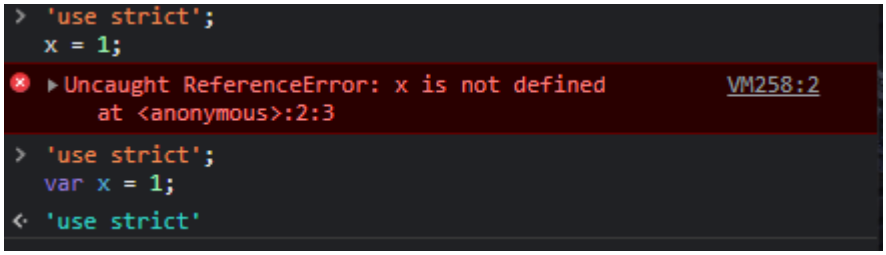
```
if (true){
  console.log(this);
}
```

if문 안에 this를 넣어도 전역적인 문맥에서 this에 접근한 것이기 때문에 윈도우 객체를 가르킨다.

그냥 외워야 한다.

```
// use strict란 엄격모드 활성화, 즉 문법을 더 엄격하게 검사하도록 하는 것이다.
'use strict';
console.log(this);
```

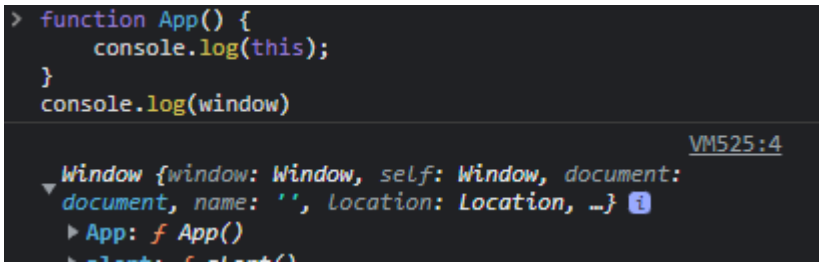




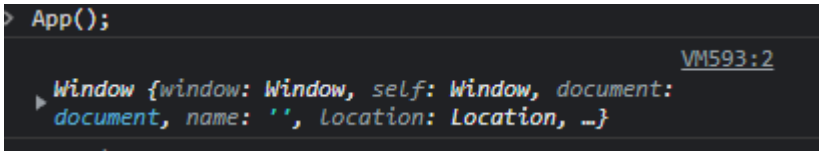
전역전인 this는 use strict와 상관없이 window 객체를 가르킨킨다.

‘this는 함수를 호출한 객체이다’

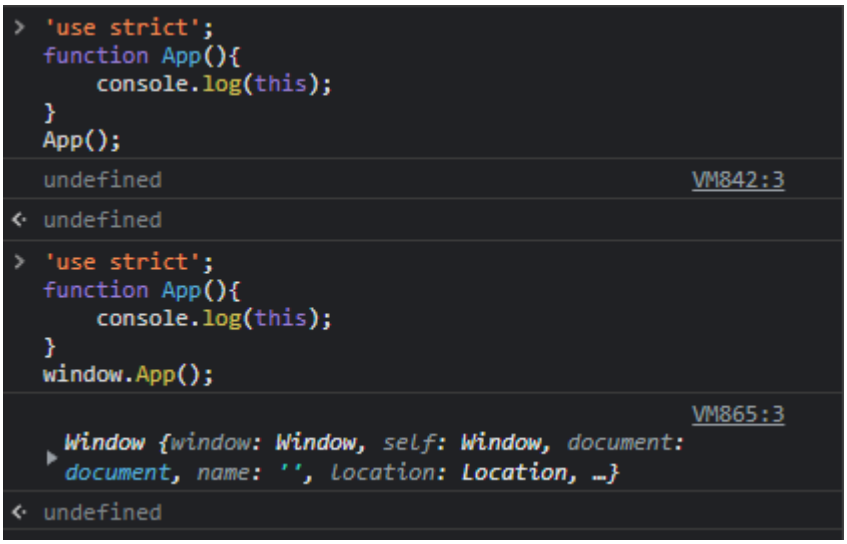
function 이라는 키워드를 사용해서 함수를 정의하게 되면 이 함수는 윈도우 객체에 등록이 된다.



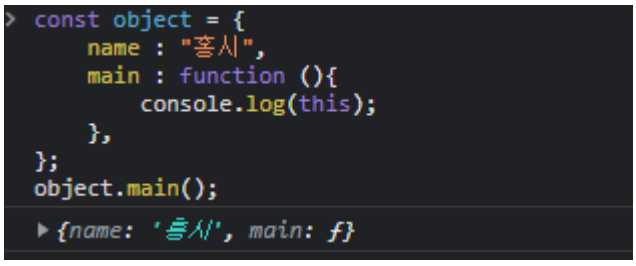
이제 App()으로 명령어를 실행하면 window가 불려지는것을 알수 있다.



이 App(); 으로 호출하는 것은 window.App();으로 호출한것과 같은 의미이다.



‘use strict’ 를 사용했을때 차이.



메서드를 만든다.

main이라는 함수를 object 객체의 메서드로 넣는다.

메서드란 객체의 속성으로 넣어진 함수를 의미한다.

main 함수 안에 this는 우리가 만든 object가 된것을 볼 수 있다.

객체의 다른 속성에 접근할 때 유용하다.

name에 접근 할 수 있다.

```
> const object = {
      name : "홍시",
      main : function (){
          console.log(this.name);
      },
  };
  object.main();
  홍시
```

함수를 호출한 객체가 중요하다.

중요한것은

this는 함수가 정의된 위치나 방법에 영향을 받지 않는다.

```
> function main(){
      console.log(this);
  }
  const object = {
      name : "홍시",
      main,
  };
  object.main();
  ▶ {name: '홍시', main: ƒ}
```

함수가 오브젝트 밖에서 정의가 되었던 안에서 정의되었던 상관없이 메인함수를 호출한 객체가 오브젝트라면 this는 오브젝트가 되는것이다.

```
> function main(){
      console.log(this);
  }
  const object = {
      name : "홍시",
      smallObject : {
          name: "작은 홍시",
          main,
      }
  };
  object.smallObject.main();
  ▶ {name: '작은 홍시', main: ƒ}
  undefined
```

smallObject 가 this값이 되었다.

쉽게 this값이 궁금하다면 main 값 . 바로 왼쪽의 값이 this 값이다.

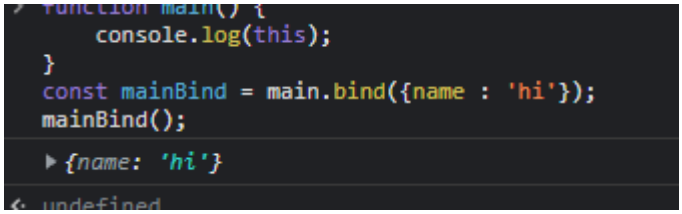
bind()

동적으로 계속 바뀌는 this값을 일일이 추적하는 것은 귀찮은 일이다.

만약 우리가 this값이 바뀌지 않도록 우리가 원하는 객체로 고정하고 싶다면 어떻게 해야할까.

함수를 누가 어떻게 호출하던 상관없이 말이다.

이때 bind를 사용하면 된다.



bind의 인자로 원하는 값을 넣는다.

this값이 window가 아니라 우리가 넣은 값이 나온다.

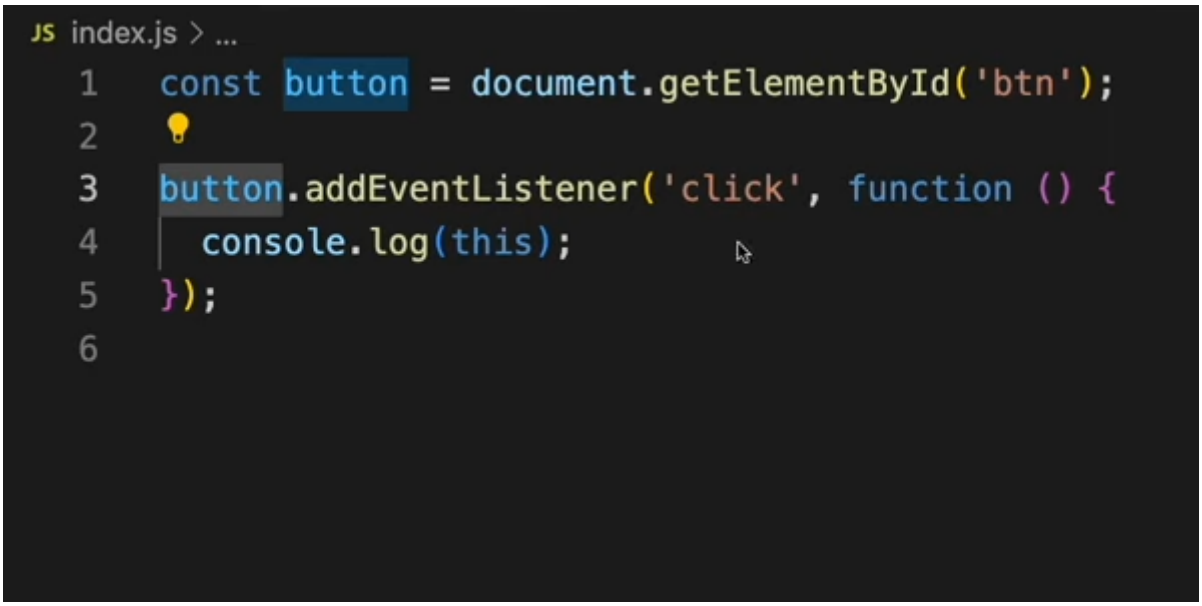
bind를 붙히면 우리가 넣은 객체가 this값으로 설정된 새로운 함수를 반환해준다.

새로운 함수가 mainbind에 들어가서 우리는 mainbind를 호출해준것이다.

주의할 점은 bind 한것은 또 bind 할 수 없다.

또 binding을 하게 되면 후에 것은 무시된다.

이벤트 처리기



여기서 this값은 click 버튼 요소가 된다.