18장

함수와 일급 객체

18.1 일급 객체

일급 객체 조건 4가지

1. 무명의 리터럴로 생성 가능.
2. 변수나 자료구조에 저장 가능.
3. 함수의 매개변수로 전달 가능.
4. 함수의 반환값으로 사용 가능.

함수는 모든 조건을
만족하므로 **일급객체**이다.

18.2 함수 객체의 프로퍼티

```
> function square(number) {
      return number*number;
  }

  console.log(Object.getOwnPropertyDescriptors(square));
```

```
▼ {length: {…}, name: {…}, arguments: {…}, caller: {…}, prototype: {…}}
  ▶ arguments: {value: null, writable: false, enumerable: false, configurable: false}
  ▶ caller: {value: null, writable: false, enumerable: false, configurable: false}
  ▶ length: {value: 1, writable: false, enumerable: false, configurable: true}
  ▶ name: {value: 'square', writable: false, enumerable: false, configurable: true}
  ▶ prototype: {value: {…}, writable: true, enumerable: false, configurable: false}
  ▶ [[Prototype]]: Object
```

```
< undefined
>
```

18.2 함수 객체의 프로퍼티

arguments 프로퍼티

```
> function multiply(x,y) {
      console.log(arguments);
      return x*y;
  }

  console.log(multiply(2,4));
```

```
▼Arguments(2) [2, 4, callee: ƒ, Symbol(Symbol.iterator): ƒ]
    0: 2
    1: 4
  ▶ callee: ƒ multiply(x,y)
    length: 2
  ▶ Symbol(Symbol.iterator): ƒ values()
  ▶ [[Prototype]]: Object
```

```
8
```

18.2 함수 객체의 프로퍼티

arguments 프로퍼티

```
> function multiply(x,y) {
      console.log(arguments);
      return x*y;
  }

  console.log(multiply(2,4,8));
```

```
▼Arguments(3) [2, 4, 8, callee: ƒ, Symbol(Symbol.iterator): ƒ]
    0: 2
    1: 4
    2: 8
  ▶ callee: ƒ multiply(x,y)
    length: 3
  ▶ Symbol(Symbol.iterator): ƒ values()
  ▶ [[Prototype]]: Object
```

```
8
```

18.2 함수 객체의 프로퍼티

arguments 프로퍼티

- arguments 객체는 **유사배열객체** 이므로 length 프로퍼티를 가짐.

- 유사배열 객체는 배열이 아니므로 **배열메소드 사용시** 오류.

- 이를 해결하기 위해 ES6에서 **Rest 파라미터** 도입.

18.2 함수 객체의 프로퍼티

caller 프로퍼티

```
> function foo(func) {
      return func();
  }

  function bar() {
      return bar.caller;
  }

  console.log(foo(bar));
  console.log(bar());
  ƒ foo(func) {
      return func();
  }
  null
<· undefined
>
```

18.2 함수 객체의 프로퍼티

length 프로퍼티

```
> function multiply(x, y) {
      return x*y;
  }

  console.log(multiply.length);
  2
```

arguments 객체의 length 프로퍼티와 **함수**의 length 프로퍼티는 다르다.

18.2 함수 객체의 프로퍼티

name 프로퍼티

```
> var namedFunc = function foo() {};
  console.log(namedFunc.name);

  foo
```

name 프로퍼티는
함수이름을 가르킨다.

```
> var anonymous = function() {};
  console.log(anonymous.name);

  anonymous
```

익명함수 표현식의 경우
식별자를 값으로 갖는다.

18.2 함수 객체의 프로퍼티

__proto__ 접근자 프로퍼티

- [[prototype]] 내부 슬롯에 __proto__ 접근자 프로퍼티를 이용해 **간접적**으로 프로토타입 객체에 접근 가능.

prototype 프로퍼티

- prototype 프로퍼티는 생성될 인스턴스의 프로토타입 객체를 가르킴.
- 생성자 함수로 호출할 수 있는 객체만 소유하는 프로퍼티.