

# Function Context and Built-in Functions

---



**PRATEEK PAREKH**

SENIOR STAFF ENGINEER

@ prparekh83



# Introduction



## Understanding context

*call* and *apply*

*bind*

## Built-in functions

- *eval*
- *parseInt*
- *parseFloat*
- *escape*
- *unescape*



# Understanding Context

---



# Function Context

```
function sayHi() {  
    console.log('Hi');  
    console.log(this);  
}  
  
sayHi(); // Hi  
        // Window {...}
```



# Function Context

```
let greeting = {};  
greeting.sayHi = function(){  
    console.log('Hi');  
    console.log(this);  
}  
greeting.sayHi(); // Hi  
                  // {sayHi:f}
```



# Function Context

```
function sayHi() {  
    console.log('Hi');  
    console.log(this);  
}  
  
let greeting = new sayHi(); // Hi  
                        // [obj Object]
```



*call()*

---



# call

```
let person1 = {name: 'John', age: 22}
```

```
let person2 = {name: 'Mary', age: 26}
```

```
let sayHi = function(){  
    console.log('Hi, ' + this.name);  
}
```

```
sayHi.call(person1) // Hi, John
```

```
sayHi.call(person2) // Hi, Mary
```





# call

```
let person1 = {name: 'John', age: 22}

let sayHi = function(message){
    console.log(message + ', ' + this.name);
}

sayHi.call(person1, 'Hi') // Hi, John
```



# Demo



*call()*



*apply()*

---



# apply

```
function introduction(name, profession) {  
    console.log('My name is ' + name + ' and I am a ' + profession + '.');  
    console.log(this);  
}  
  
introduction('John', 'student');  
  
introduction.apply(undefined, ['Mary', 'Lawyer']);  
  
introduction.call(undefined, 'James', 'artist');
```



# apply vs call

**apply()**

**array input  
with similar elements**

**call()**

**individual arguments  
of varying type**



# Demo



*apply()*



*bind()*

---



# bind

```
let person1 = {name: 'Mary',  
  getName: function(){  
    return this.name;  
  }  
};  
  
let person2 = {name: 'John'};  
  
let getNameCopy = person1.getName.bind(person2);  
  
console.log(getNameCopy());    // John
```





# Demo



*bind()*



# Built-in functions

---



# eval

```
let x = 1;
```

```
let y = 2;
```

```
console.log(eval('x + y + 1')); // 4
```



# eval

```
let x = 1;
```

```
let y = 2;
```

```
let s = 'abc';
```

```
console.log(eval('x + y + s')); // 3abc
```



# parseInt

```
console.log(parseInt('F', 16)); // 15  
console.log(parseInt('15', 10)); // 15  
console.log(parseInt('Hi', 10)); // NaN
```



# parseFloat

```
console.log(parseFloat('3.99')); // 3.99
```

```
console.log(parseFloat('3.99e-1')); // 39.9
```

```
console.log(parseFloat('')); // NaN
```



# escape

```
console.log(escape('text')); // text  
console.log(escape(' ')); // %20  
console.log(escape('abc&%')); // abc%26%25
```



# unescape

```
console.log(unescape('text')); // text
```

```
console.log(unescape('%20')); //
```

```
console.log(unescape('abc%26%25')); // abc&%
```





# Demo



## Built-in Functions



# Summary



*apply* and *call*

*bind*

Built-in functions

