

# Promises, Iterators, and Generators

Level 6

# Generators

Level 6 – Section 3



# Generator Functions

The *function* \* declaration defines **generator functions**. These are special functions from which we can use the *yield* keyword to return **iterator** objects.

It's a star character!

```
function *nameList(){  
  yield "Sam";  
  yield "Tyler";  
}
```

→ { done: false, value: "Sam" }  
→ { done: false, value: "Tyler" }

```
function *nameList(){  
function *nameList(){  
function *nameList(){
```

Doesn't matter where we place the star, as long as it's the first thing after the function keyword

# Generator Objects and for...of

Generator functions return objects that provide the same *next* method expected by *for...of*, the *spread operator*, and the *destructuring assignment*.

```
function *nameList(){  
  yield "Sam";  
  yield "Tyler";  
}
```

→ { done: false, value: "Sam" }  
→ { done: false, value: "Tyler" }

Calling the function returns a generator object

```
for(let name of nameList()){  
  console.log( name );  
}
```

> Sam  
> Tyler

```
let names = [...nameList()];  
console.log( names );
```

> ["Sam", "Tyler"]

```
let [first, second] = nameList();  
console.log( first, second );
```

> Sam Tyler



# Replacing Manual Iterator Objects

Knowing how to **manually** craft an iterator object is important, but there's a **shorter** syntax.

```
let post = { title: "New Features in JS", replies: 19 };

post[Symbol.iterator] = function(){

  let properties = Object.keys(this);
  let count = 0;
  let isDone = false;

  let next = () => {
    if(count >= properties.length){
      isDone = true;
    }
    return { done: isDone, value: this[properties[count++]] };
  }
  return { next };
}
```

We can make this shorter



# Refactoring to Generator Functions

Each time *yield* is called, our function returns a **new iterator** object and then **pauses** until it's called again.

```
let post = { title: "New Features in JS", replies: 19 };
```

```
post[Symbol.iterator] = function *(){
```

← Generator function signature

```
  let properties = Object.keys(this);  
  for(let p of properties){  
    yield this[p];  
  }  
}
```

first 'title'  
then 'replies'

Same as manually returning each property value

```
post[Symbol.iterator] = function *(){  
  yield this.title;  
  yield this.replies;  
}
```

```
for(let p of post){  
  console.log( p );  
}
```

Successfully returns correct values!

> New Features in JS  
> 19

