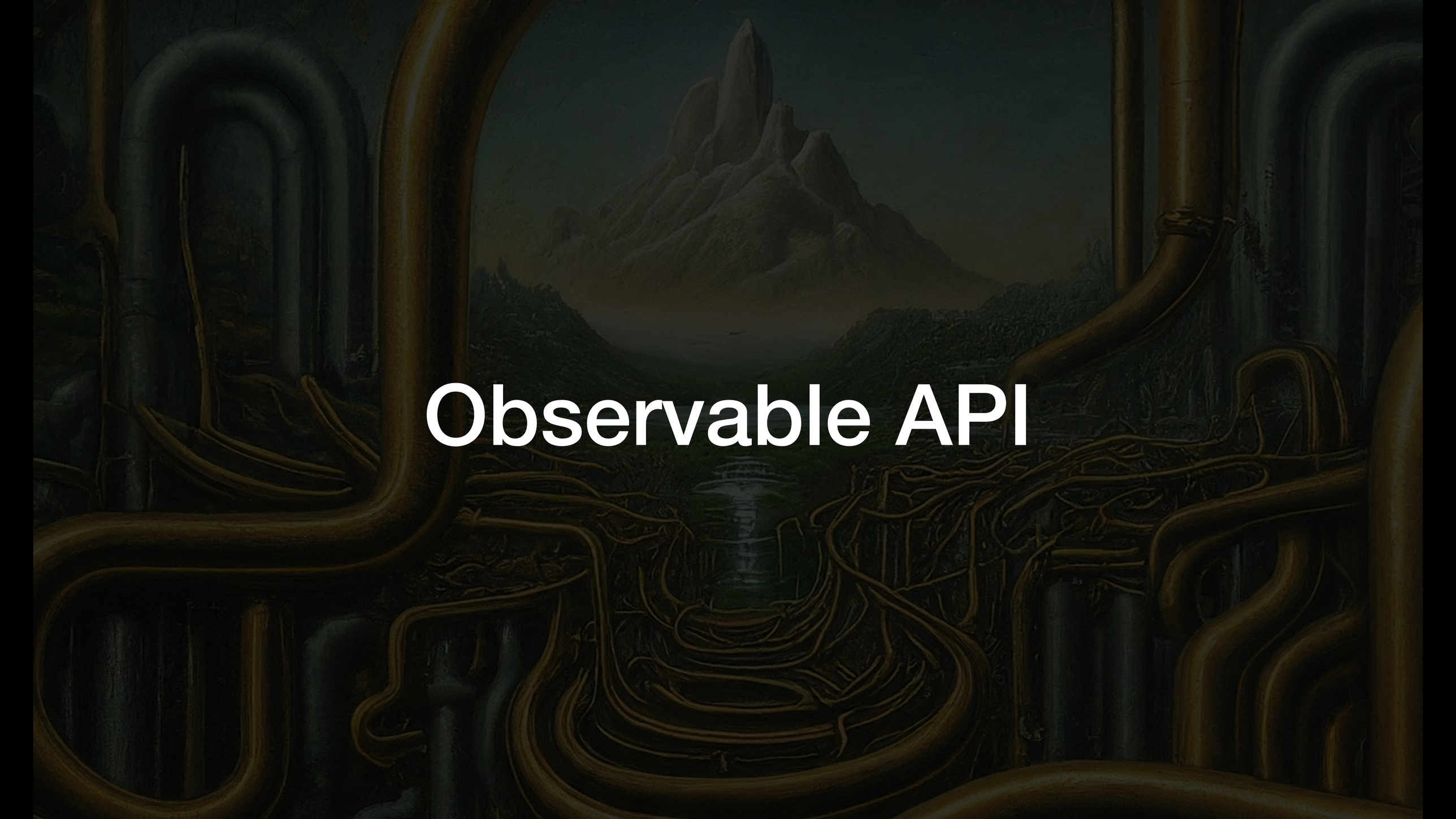
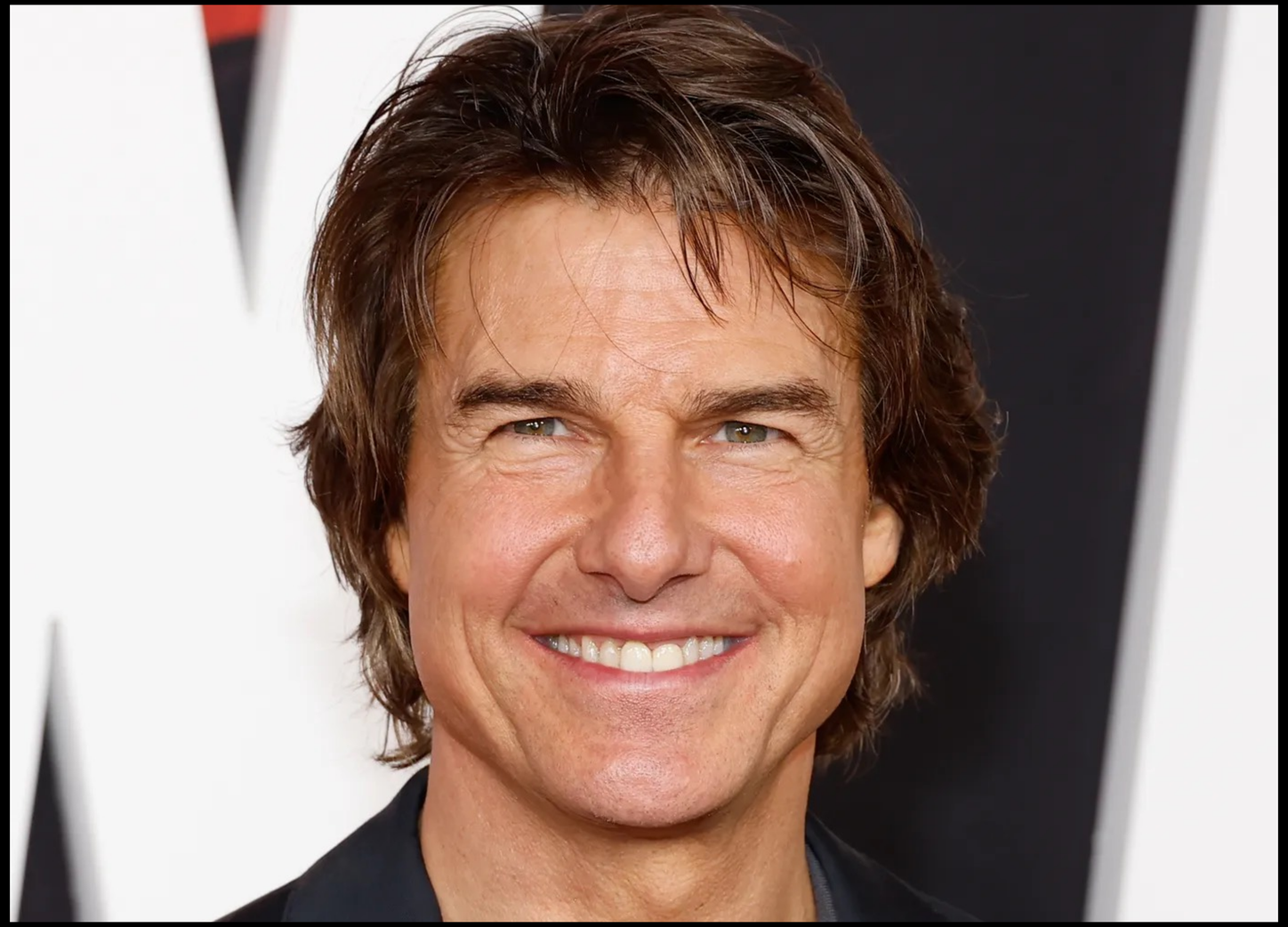

Observable API

Dominic Farolino

- Google Chrome Engineer
- WHATWG Editor

“Don”



“Tom”

Dom Farolino

- Google Chrome Engineer
- WHATWG Editor

twitter.com/domfarolino

github.com/domfarolino

What is this talk?

dom@chromium.org

History of async on the Web

(this could really be its own talk)

@domfarolino

dom@chromium.org

What's an Observable?

dom@chromium.org

Observables in action

dom@chromium.org

Observable operators

dom@chromium.org

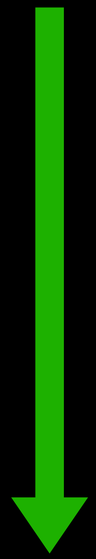
Unsubscription & teardown

dom@chromium.org

Promises & callbacks

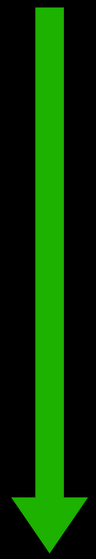
@domfarolino

dom@chromium.org



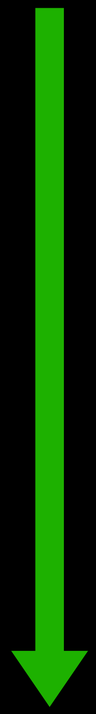
```
// some code

const result = doLotsOfSyncStuff();

// use `result`…
```


dom@chromium.org



```
// some code

doAsyncStuff();

// more code?…
```


dom@chromium.org



```
// some code

doAsyncStuff(function doStuff(result) {
  //
  // use `result`
  //
});

// more code?…
```


dom@chromium.org

```
// some code

doAsyncStuff(function doStuff(result) {
  // …
  // use `result`
  // …
});

// more code…
```


dom@chromium.org

Non-linear control flow

dom@chromium.org

```
function hell(win) {
 // for listener purpose
 return function() {
   loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
     loadLink(win, REMOTE_SRC+'/lib/async.js', function() {
       loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {
         loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {
           loadLink(win, REMOTE_SRC+'/lib/underscode.min.js', function() {
             loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {
               loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {
                 loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {
                   loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {
                     async.eachSeries(SCRIPTS, function(src, callback) {
                       loadScript(win, BASE_URL+src, callback);
                     });
                   });
                 });
               });
             });
           });
         });
       });
     });
   });
 };
}
```

https://dev.to/jerrycode06/callback-hell-and-how-to-rescue-it-ggj

```
function hell(win) {
 // for listener purpose
 return function() {
   loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {
     loadLink(win, REMOTE_SRC+'/lib/async.js', function() {
       loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {
         loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {
           loadLink(win, REMOTE_SRC+'/lib/underscode.min.js', function() {
             loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {
               loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {
                 loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {
                   loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {
                     async.eachSeries(SCRIPTS, function(src, callback) {
                       loadScript(win, BASE_URL+src, callback);
                     });
                   });
                 });
               });
             });
           });
         });
       });
     });
   });
 };
}
```

https://dev.to/jerrycode06/callback-hell-and-how-to-rescue-it-ggj

@domfarolino

dom@chromium.org

Synchronously available
handle for async work

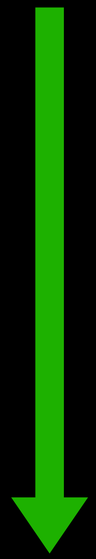
dom@chromium.org

Promises

ECMAScript 2015

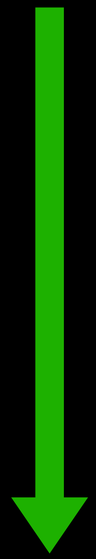
@domfarolino

dom@chromium.org



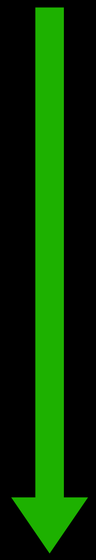
```
// some code

const p = doAsyncStuff();

p.then(v => {
  // more code?…
});
```


dom@chromium.org



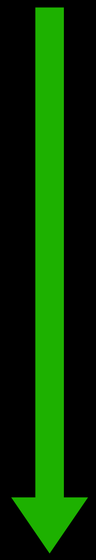
```
// some code

const p = doAsyncStuff();

p.then(v => {
  // more code?…
});
```


dom@chromium.org



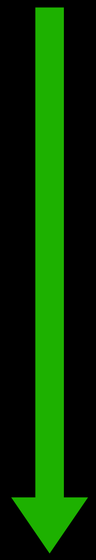
```
// some code

const p = doAsyncStuff();

p.then(v => {
  // more code?…
});
```


dom@chromium.org



```
p.then(v => {
  // more code?…
});
```


dom@chromium.org



```
p.then(v => {
  // more code…
}).then(v => {
  // more code…
}).then(v => {
  // more code…
});
```


dom@chromium.org

Promise limitations

dom@chromium.org

```
const p = new Promise(resolve => {
  element.onclick = resolve;
});
```


dom@chromium.org

```
const p = new Promise(resolve => {
  element.onclick = resolve;
});
```


dom@chromium.org

Forced scheduling

dom@chromium.org

```
const p = new Promise(resolve => {
  element.onclick = resolve;
});
```


dom@chromium.org

```
const p = new Promise(resolve => {
  element.onclick = resolve;
});
```


dom@chromium.org

```
const p = new Promise(resolve => {
  element.onclick = resolve;
});

p.then(event => {
  // Do something with `event`…
});
```


dom@chromium.org

```
const p = new Promise(resolve => {
  element.onclick = resolve;
});

p.then(event => {
  // Do something with `event`…
});
```


dom@chromium.org

```
const p = new Promise(resolve => {
  element.onclick = resolve;
});

p.then(event => {
⌛// Do something with `event`…
});
```


dom@chromium.org

```
const p = new Promise(resolve => {
  element.onclick = resolve;
});

p.then(event => {
  event.preventDefault();
});
```


dom@chromium.org

Promise

EventTarget

dom@chromium.org

Promise

EventTarget

dom@chromium.org

```
interface EventTarget {
  constructor();

  undefined addEventListener(DOMString type, EventListener? callback,   ...);
  undefined removeEventListener(DOMString type, EventListener? callback, ...  ) );
  boolean dispatchEvent(Event event);
};
```

https://dom.spec.whatwg.org

```
interface EventTarget {
  constructor();

  undefined addEventListener(DOMString type, EventListener? callback,   ...);
  undefined removeEventListener(DOMString type, EventListener? callback, ...  ) );
  boolean dispatchEvent(Event event);
};
```

https://dom.spec.whatwg.org

`addEventListener()` hell

dom@chromium.org

Konami example

dom@chromium.org

```
const pattern = [
  'ArrowUp',
  'ArrowUp',
  'ArrowDown',
  'ArrowDown',
  'ArrowLeft',
  'ArrowRight',
  'ArrowLeft',
  'ArrowRight',
  'b',
  'a',
  'b',
  'a',
  'Enter',
];
```


dom@chromium.org

```
// Imperative
document.addEventListener('keydown', e => {
  const key = e.key;
  if (key === pattern[0]) {
    let i = 1;
    const handler = (e) => {
      const nextKey = e.key;
      if (nextKey !== pattern[i++]) {
        document.removeEventListener('keydown', handler)
      } else if (pattern.length === i) {
        console.log('Secret code matched!');
        document.removeEventListener('keydown', handler)
      }
    };

    document.addEventListener('keydown', handler);
  }
}, {once: true});
```


dom@chromium.org

```
// Imperative
document.addEventListener('keydown', e => {
  const key = e.key;
  if (key === pattern[0]) {
    let i = 1;
    const handler = (e) => {
      const nextKey = e.key;
      if (nextKey !== pattern[i++]) {
        document.removeEventListener('keydown', handler)
      } else if (pattern.length === i) {
        console.log('Secret code matched!');
        document.removeEventListener('keydown', handler)
      }
    };

    document.addEventListener('keydown', handler);
  }
}, {once: true});
```


dom@chromium.org

```
// Imperative
document.addEventListener('keydown', e => {
  const key = e.key;
  if (key === pattern[0]) {
    let i = 1;
    const handler = (e) => {
      const nextKey = e.key;
      if (nextKey !== pattern[i++]) {
        document.removeEventListener('keydown', handler)
      } else if (pattern.length === i) {
        console.log('Secret code matched!');
        document.removeEventListener('keydown', handler)
      }
    };

    document.addEventListener('keydown', handler);
  }
}, {once: true});
```


dom@chromium.org

```
// Imperative
document.addEventListener('keydown', e => {
  const key = e.key;
  if (key === pattern[0]) {
    let i = 1;
    const handler = (e) => {
      const nextKey = e.key;
      if (nextKey !== pattern[i++]) {
        document.removeEventListener('keydown', handler)
      } else if (pattern.length === i) {
        console.log('Secret code matched!');
        document.removeEventListener('keydown', handler)
      }
    };

    document.addEventListener('keydown', handler);
  }
}, {once: true});
```


dom@chromium.org

```
// Imperative
document.addEventListener('keydown', e => {
  const key = e.key;
  if (key === pattern[0]) {
    let i = 1;
    const handler = (e) => {
      const nextKey = e.key;
      if (nextKey !== pattern[i++]) {
        document.removeEventListener('keydown', handler)
      } else if (pattern.length === i) {
        console.log('Secret code matched!');
        document.removeEventListener('keydown', handler)
      }
    };

    document.addEventListener('keydown', handler);
  }
}, {once: true});
```


dom@chromium.org

```
// Imperative
document.addEventListener('keydown', e => {
  const key = e.key;
  if (key === pattern[0]) {
    let i = 1;
    const handler = (e) => {
      const nextKey = e.key;
      if (nextKey !== pattern[i++]) {
        document.removeEventListener('keydown', handler)
      } else if (pattern.length === i) {
        console.log('Secret code matched!');
        document.removeEventListener('keydown', handler)
      }
    };

    document.addEventListener('keydown', handler);
  }
}, {once: true});
```


dom@chromium.org

Observables will save the day

dom@chromium.org

What's an Observable?

dom@chromium.org

Promises for multiple values

dom@chromium.org

Promises for multiple values

(sorta…)

dom@chromium.org

Observable API

1. Handle for a stream of async events
2. Manipulate the stream with operators

dom@chromium.org

Observable

EventTarget

dom@chromium.org

```
partial interface EventTarget {
  Observable on(DOMString type, optional ObservableEventListenerOptions options = {});
};
```


dom@chromium.org

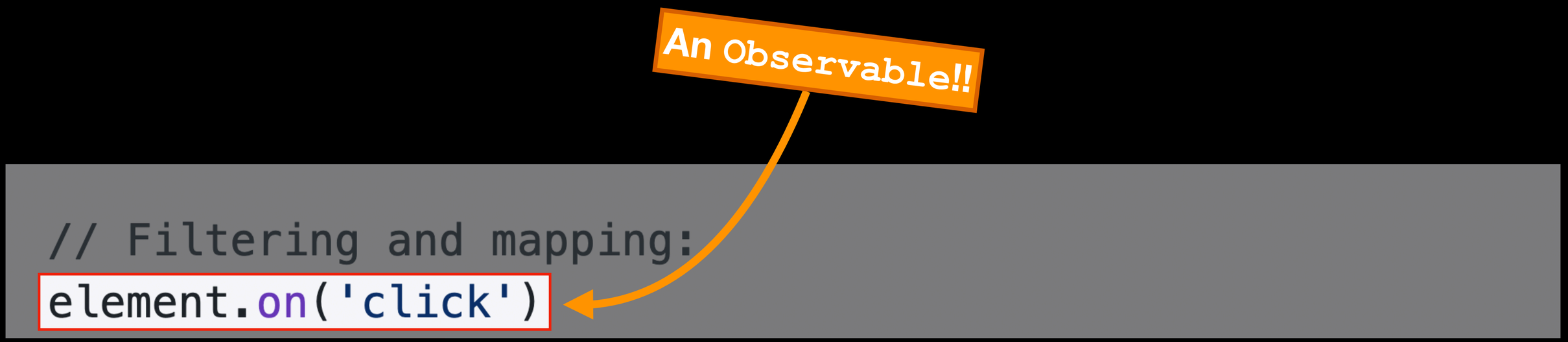
```
partial interface EventTarget {
  Observable on(DOMString type, optional ObservableEventListenerOptions options = {});
};
```

```
partial interface EventTarget {
  Observable on(DOMString type, optional ObservableEventListenerOptions options = {});
};
```

“Better `addEventListener()`”

```
// Filtering and mapping:
element.on('click')
```


dom@chromium.org



dom@chromium.org


```
// Filtering and mapping:
element.on('click')
       .filter((e) => e.target.matches('.foo'))
```

An Observable!!

```
// Filtering and mapping:
element.on('click')
       .filter((e) => e.target.matches('.foo'))
```


dom@chromium.org

An Observable!!

```
// Filtering and mapping:
element.on('click')
       .filter((e) => e.target.matches('.foo'))
       .map((e) => ({ x: e.clientX, y: e.clientY }))
```


dom@chromium.org

An Observable!!

```
// Filtering and mapping:
element.on('click')
       .filter((e) => e.target.matches('.foo'))
       .map((e) => ({ x: e.clientX, y: e.clientY }))
```


dom@chromium.org

```
// Filtering and mapping:
element.on('click')
       .filter((e) => e.target.matches('.foo'))
       .map((e) => ({ x: e.clientX, y: e.clientY }))
       .subscribe(event => handleClickAtPoint(event));
```

```
// Filtering and mapping:
element.on('click')
        .filter((e) => e.target.matches('.foo'))
        .map((e) => ({ x: e.clientX, y: e.clientY }))
        .subscribe(event => handleClickAtPoint(event));
```

```
// Filtering and mapping:
element.on('click')
       .filter((e) => e.target.matches('.foo'))
       .map((e) => ({ x: e.clientX, y: e.clientY }))
       .subscribe(event => handleClickAtPoint(event));
```

Event

dom@chromium.org

```
// Filtering and mapping:
element.on('click')
        .filter( Event => e.target.matches('.foo'))
        .map((e) => ({ x: e.clientX, y: e.clientY }))
        .subscribe(event => handleClickAtPoint(event));
```


dom@chromium.org

```
// Filtering and mapping:
element.on('click')
        .filter((e) => e.target.matches('.foo'))
        .map((e) => ({ x: e.clientX, y: e.clientY }))
        .subscribe(event => handleClickAtPoint(event));
```

Event

dom@chromium.org

```
// Filtering and mapping:
element.on('click')
       .filter((e) => e.target.matches('.foo'))
       .map((e {x:80 y:76} x: e.clientX, y: e.clientY }))
       .subscribe(event => handleClickAtPoint(event));
```


dom@chromium.org

```
// Filtering and mapping:
element.on('click')
        .filter((e) => e.target.matches('.foo'))
        .map((e) => ({ x: e.clientX, y: e.clientY }))
        .subscribe({x:80 y:76}> handleClickAtPoint(event));
```


dom@chromium.org

Konami example

dom@chromium.org

```
const keys = document.on('keydown')
```


dom@chromium.org

```
const keys = document.on('keydown')
  .map(e => e.key)
```


dom@chromium.org

```
const keys = document.on('keydown')
  .map(e => e.key)
  .take(pattern.length)
```


dom@chromium.org

```
const keys = document.on('keydown')
  .map(e => e.key)
  .take(pattern.length)
  .every((k, i) => k === pattern[i])
```


dom@chromium.org

```
const keys = document.on('keydown')
  .map(e => e.key)
  .take(pattern.length)
  .every((k, i) => k === pattern[i])
  .then(matched => console.log(matched ? 'Success!' : 'Failure'));
```


dom@chromium.org

Making your own Observable

dom@chromium.org

“Like a” Promise

dom@chromium.org

Producer & consumer

```
const p = new Promise(resolve => {
  resolve(10);
});
```


dom@chromium.org

Producer & consumer

```
const p = new Promise(resolve => {
  resolve(10);
});

p.then(v => {
  console.log(v);
});
```


dom@chromium.org

Producer & consumer

```
const source = new Observable(
  // Producer code…
);
```


dom@chromium.org

Producer & consumer

```
const source = new Observable(subscriber => {
  // Producer code…
});
```


dom@chromium.org

Producer & consumer

```
const source = new Observable(subscriber => {
  subscriber.next(10);
});
```


dom@chromium.org

Producer & consumer

```
const source = new Observable(subscriber => {
  subscriber.next(10);
});

source.subscribe(v => {
  console.log(v);
});
```


dom@chromium.org

Producer & consumer

```
const source = new Observable(subscriber => {
  subscriber.next(10);
});

source.subscribe(v => {
  console.log(v);
});
```


dom@chromium.org

Subscriber interface

§ 2.1. The Subscriber interface

```
[Exposed=*]
interface Subscriber {
  undefined next(any value);
  undefined error(any error);
  undefined complete();
  undefined addTeardown(VoidFunction teardown);

  // True after the Subscriber is created, up until either
  // complete()/error() are invoked, or the subscriber unsubscribes. Inside
  // complete()/error(), this attribute is true.
  readonly attribute boolean active;

  readonly attribute AbortSignal signal;
};
```

https://wicg.github.io/observable/#subscriber-api

dom@chromium.org

Subscriber interface

§ 2.1. The Subscriber interface

```
[Exposed=*]
interface Subscriber {
  undefined next(any value);
  undefined error(any error);
  undefined complete();
  undefined addTeardown(VoidFunction teardown);

  // True after the Subscriber is created, up until either
  // complete()/error() are invoked, or the subscriber unsubscribes. Inside
  // complete()/error(), this attribute is true.
  readonly attribute boolean active;

  readonly attribute AbortSignal signal;
};
```

like `resolve()`

https://wicg.github.io/observable/#subscriber-api

@domfarolino

dom@chromium.org

Subscriber interface

§ 2.1. The Subscriber interface

```
[Exposed=*]
interface Subscriber {
  undefined next(any value);
  undefined error(any error);
  undefined complete();
  undefined addTeardown(VoidFunction teardown);

  // True after the Subscriber is created, up until either
  // complete()/error() are invoked, or the subscriber unsubscribes. Inside
  // complete()/error(), this attribute is true.
  readonly attribute boolean active;

  readonly attribute AbortSignal signal;
};
```

like `reject()`

https://wicg.github.io/observable/#subscriber-api

@domfarolino

dom@chromium.org

`Subscriber` interface

§ 2.1. The `Subscriber` interface

```
[Exposed=*]
interface Subscriber {
  undefined next(any value);
  undefined error(any error);
  undefined complete();
  undefined addTeardown(VoidFunction teardown);

  // True after the Subscriber is created, up until either
  // complete()/error() are invoked, or the subscriber unsubscribes. Inside
  // complete()/error(), this attribute is true.
  readonly attribute boolean active;

  readonly attribute AbortSignal signal;
};
```

https://wicg.github.io/observable/#subscriber-api

@domfarolino

dom@chromium.org

Producer & consumer

```
const source = new Observable(subscriber => {
  subscriber.next(10);
});

source.subscribe({
  next: v => console.log(v),
  error: e => console.warn(e),
  complete: () => {},
});
```


dom@chromium.org

Producer & consumer

```
const source = new Observable(subscriber => {
  subscriber.error(new Error());
});

source.subscribe({
  next: v => console.log(v),
  error: e => console.warn(e),
  complete: () => {},
});
```


dom@chromium.org

Producer & consumer

```
const source = new Observable(subscriber => {
  subscriber.complete();
});

source.subscribe({
  next: v => console.log(v),
  error: e => console.warn(e),
  complete: () => {},
});
```


dom@chromium.org

Producer & consumer

```
const source = new Observable(subscriber => {
  subscriber.next(10);
});

source.subscribe({
  next: v => console.log(v),
  error: e => console.warn(e),
  complete: () => {},
});
```


dom@chromium.org

@domfarolino

dom@chromium.org

Lazy

dom@chromium.org

Promises: Eager

```
const p = new Promise(resolve => {
  resolve(10);
});
```


dom@chromium.org

Promises: Eager

```
const p = new Promise(resolve => {
  resolve(10);
});
```


dom@chromium.org

Observables: Lazy

```
const source = new Observable(subscriber => {
  subscriber.next(10);
});
```


dom@chromium.org

Observables: Lazy

```
const source = new Observable(subscriber => {
  subscriber.next(10);
});

source.subscribe();
```


dom@chromium.org

Observables: Lazy

```
const source = new Observable(subscriber => {
  subscriber.next(10);
});

source.subscribe();
source.subscribe();
```


dom@chromium.org

Observables: Lazy

```
const source = new Observable(subscriber => {
  subscriber.next(10);
});

source.subscribe();
source.subscribe();
source.subscribe();
```


dom@chromium.org

	Observable	Promise
# values	Many	One
Producer	Lazy	Eager
Multicast	✅	❌

dom@chromium.org

HTML

JavaScript

dom@chromium.org

HTML

```
<button id=button></button>
```

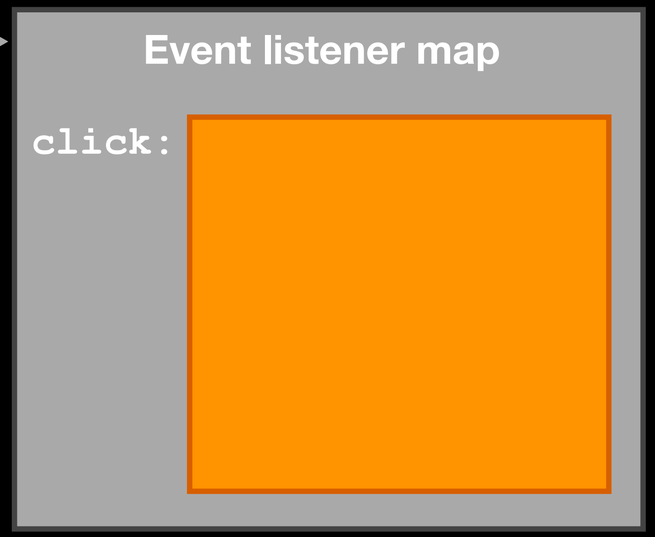
JavaScript

dom@chromium.org

HTML

JavaScript

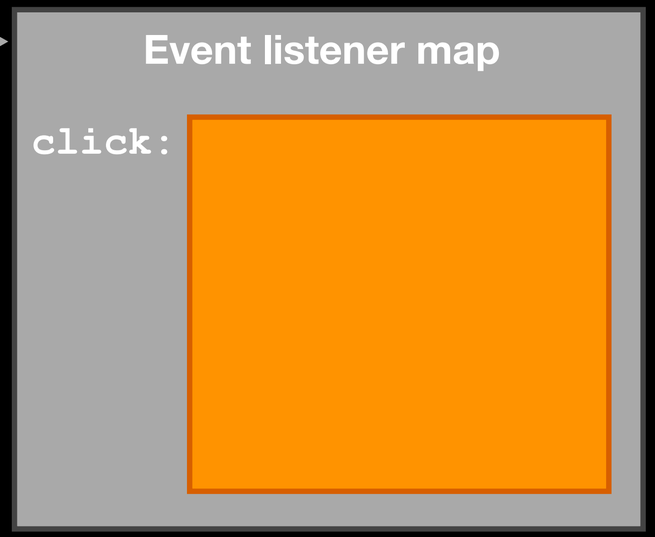
```
<button id=button></button>
```



dom@chromium.org

HTML

```
<button id=button></button>
```



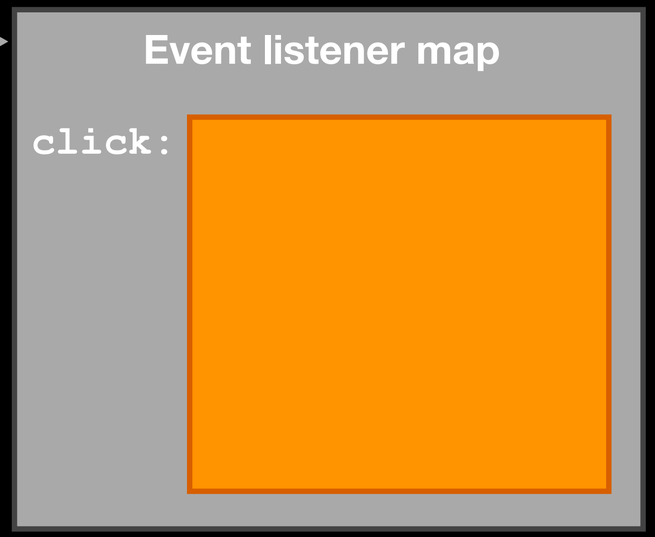
JavaScript

```
const source = button.on('click');
```


dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

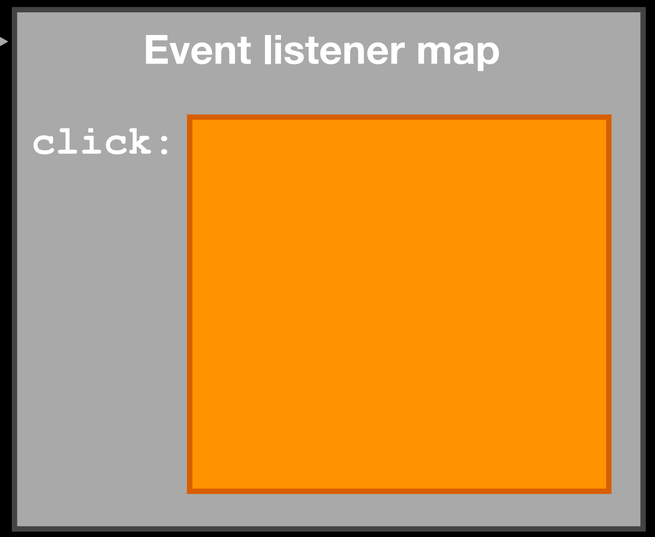
```
const source = button.on('click');
```

Native code

dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

```
const source = button.on('click');
```

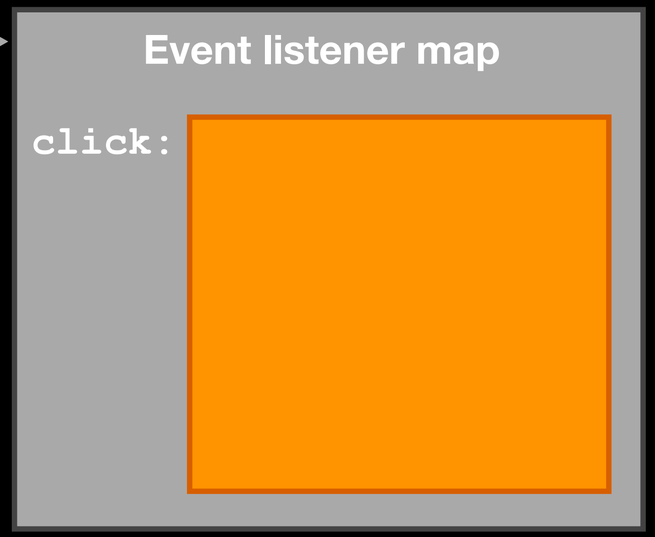
Native code

@domfarolino

dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

```
const source = button.on('click');
```

Native code

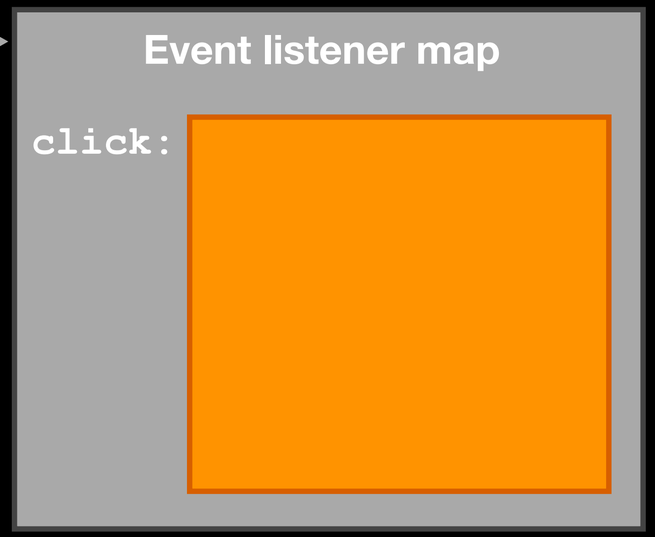
Observable

```
function onSubscribe(subscriber) {


}
```


dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

```
const source = button.on('click');
```

Native code

Observable

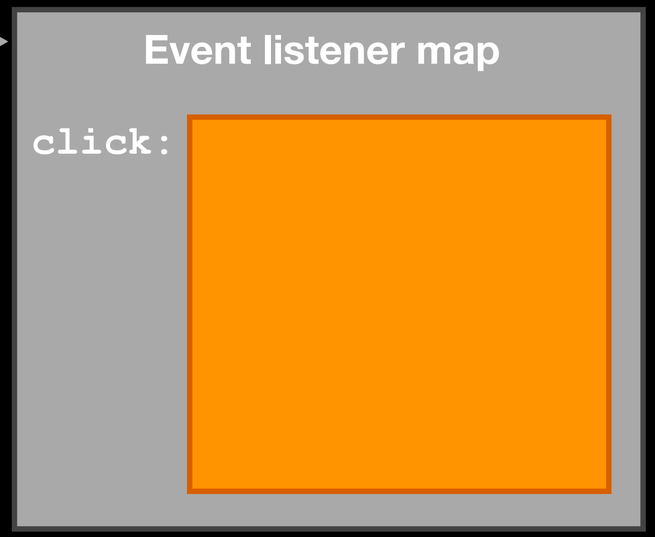
```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

```
const source = button.on('click');
```

Native code

Observable

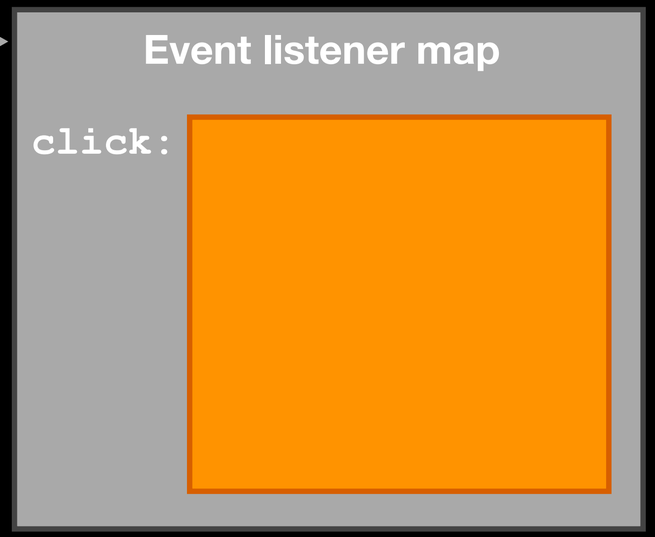
```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

```
const source = button.on('click');
```

Native code

Observable

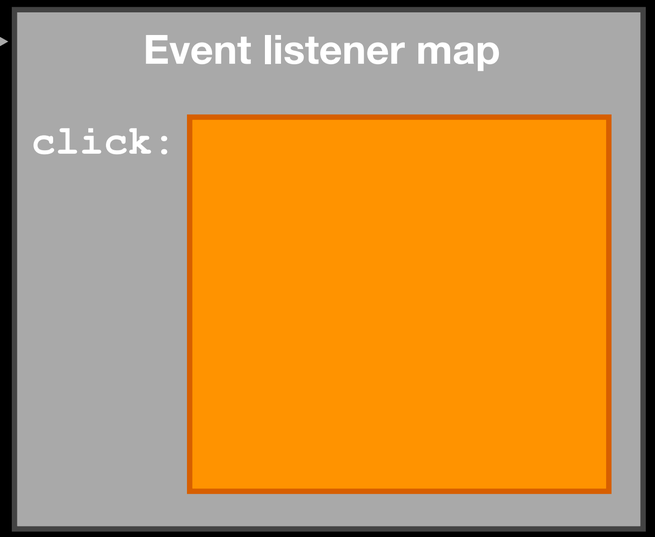
```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Native code

Observable

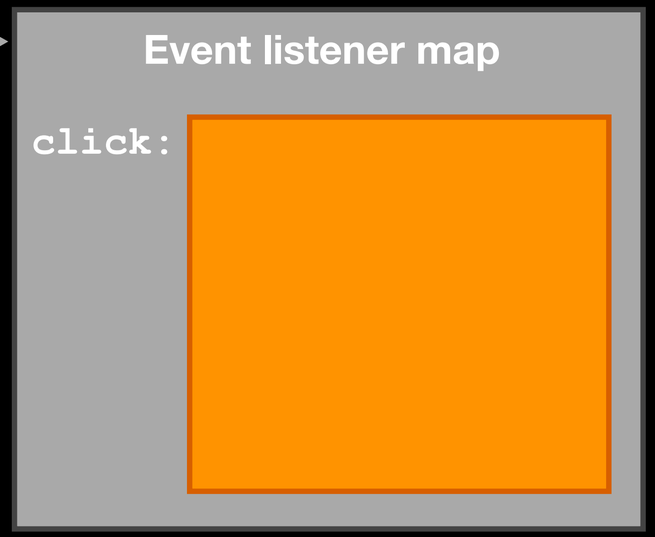
```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Native code

Observable

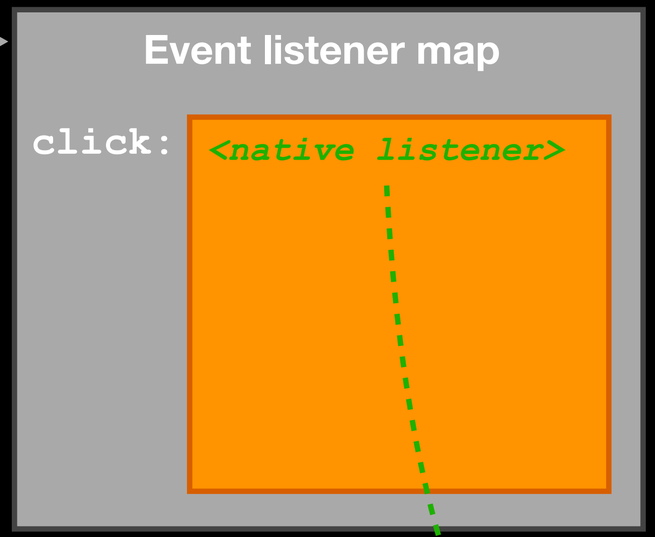
```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Native code

Observable

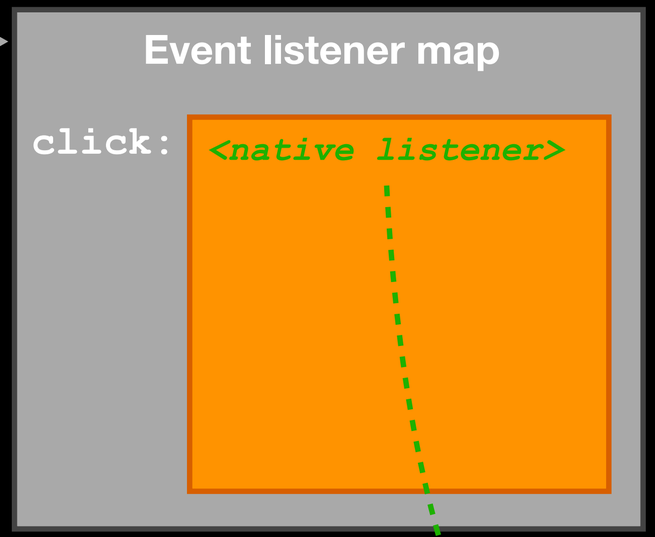
```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML

```
<button id=button></button>
```



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Native code

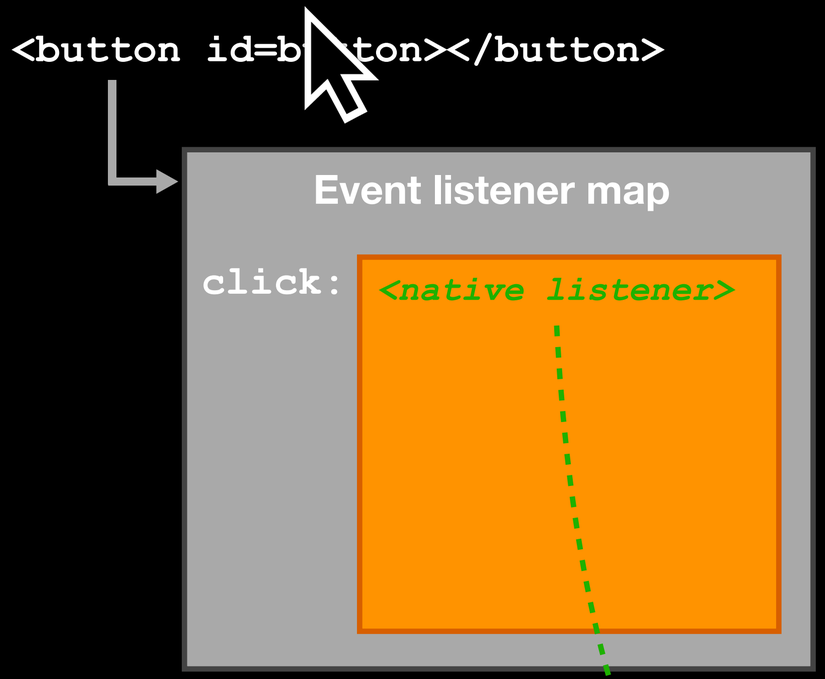
Observable

```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Native code

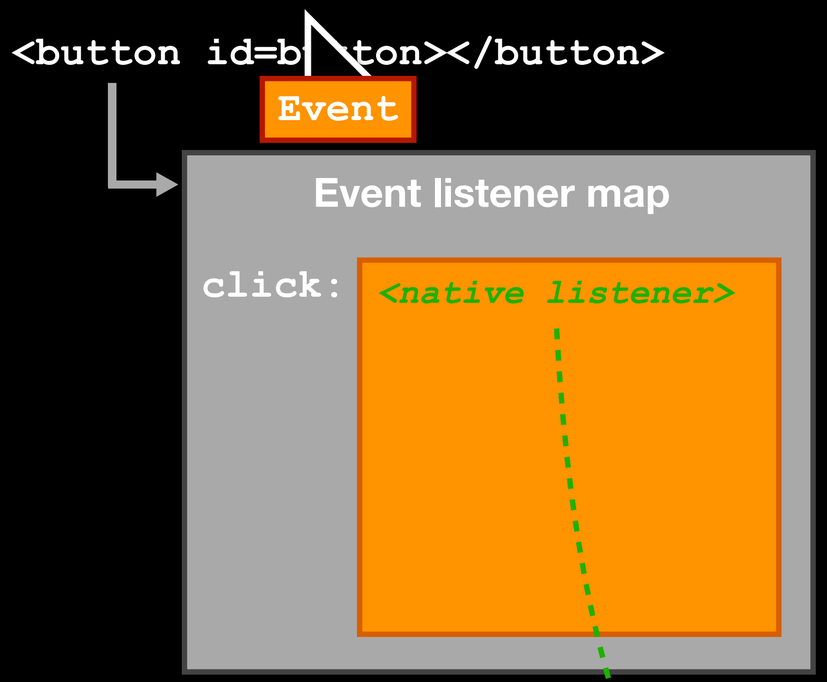
Observable

```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Native code

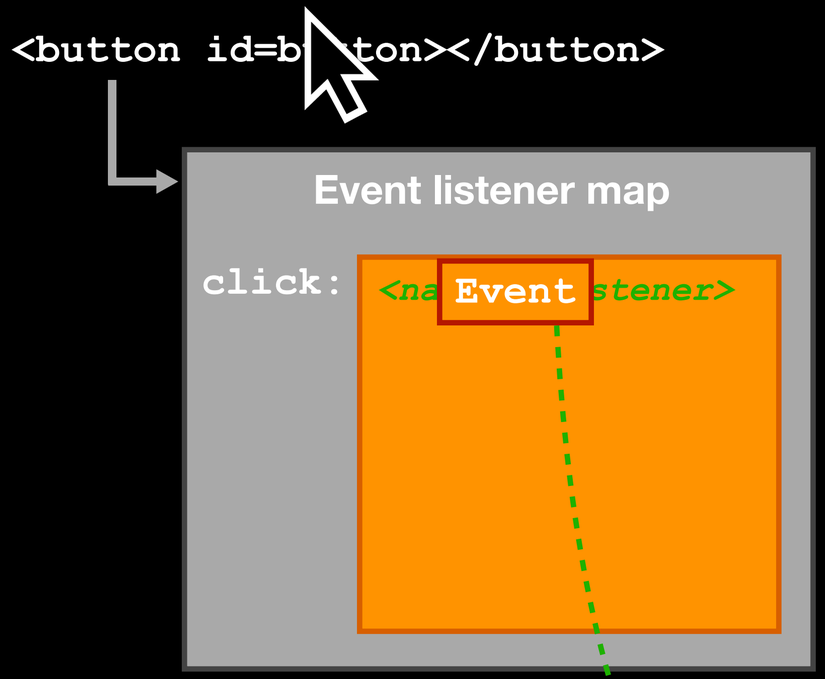
Observable

```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Native code

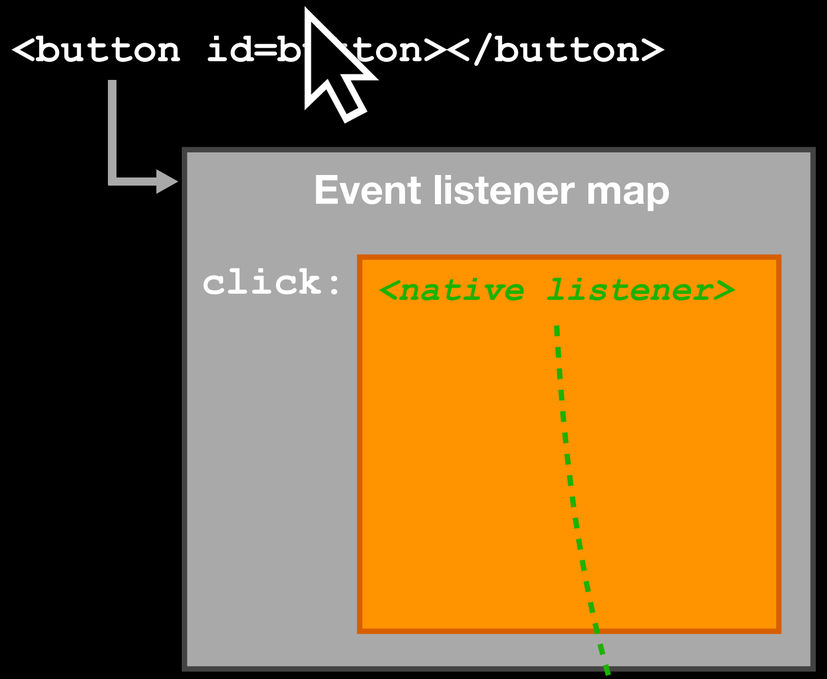
Observable

```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Native code

Observable

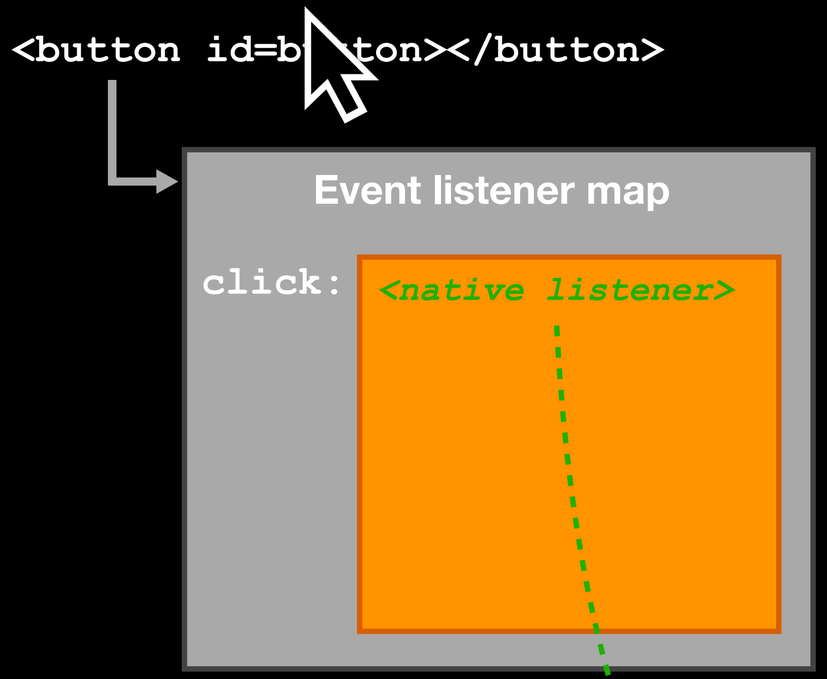
```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

Event

@domfarolino

dom@chromium.org

HTML



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Native code

Observable

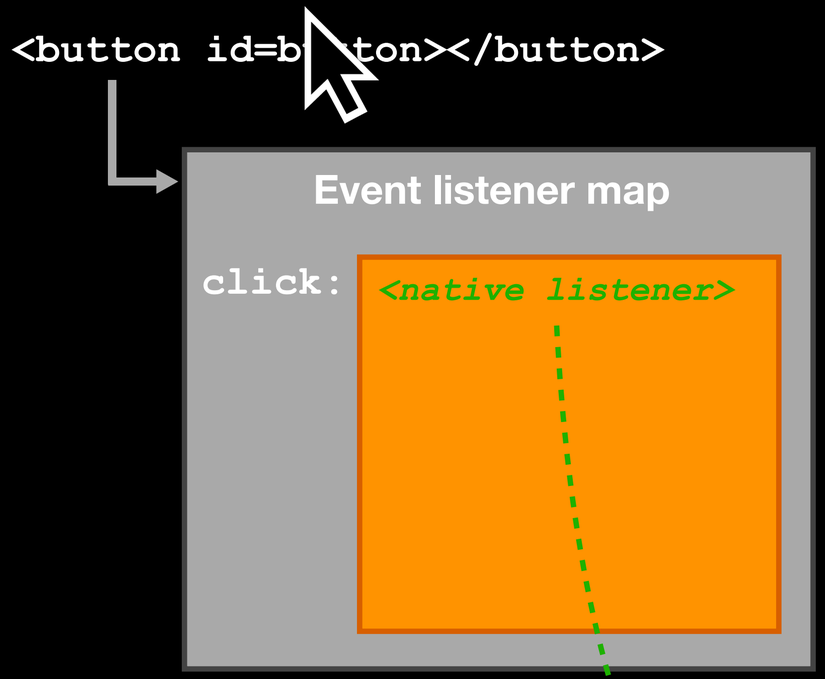
```
function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

Event

@domfarolino

dom@chromium.org

HTML



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log(e);
  },
});
```

Event

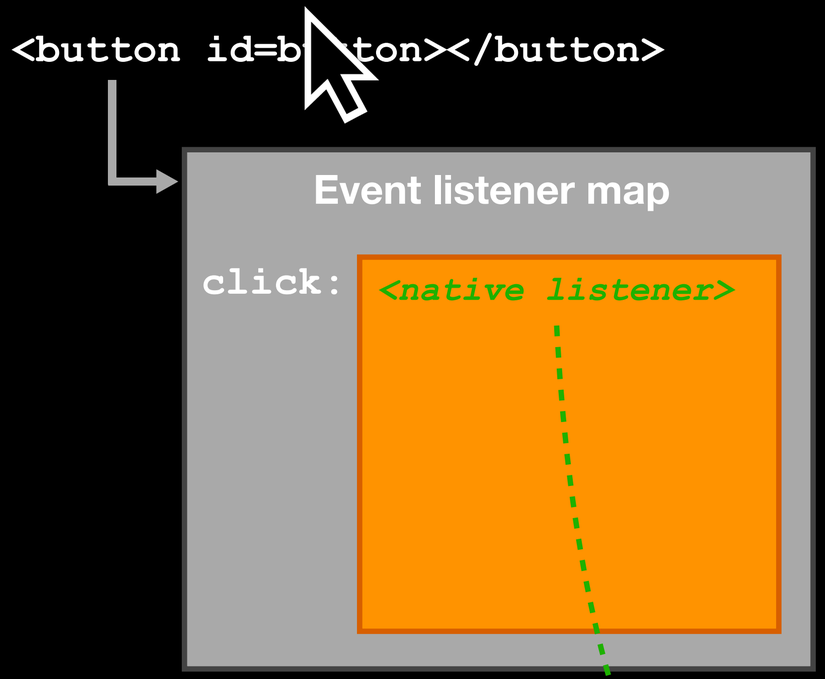
Native code

```
Observable

function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

HTML



JavaScript

```
const source = button.on('click');

source.subscribe({
  next: e => {
    console.log( Event
  },
});
```

Native code

```
Observable

function onSubscribe(subscriber) {
  button.addEventListener('click', e => {
    subscriber.next(e);
  });
}
```

@domfarolino

dom@chromium.org

	Observable	Promise
# values	Many	One
Producer	Lazy	Eager
Multicast	✅	❌
Scheduling	Sync	Async

dom@chromium.org

Observable operators

dom@chromium.org

Observable operators

`filter()`, `map()`, etc.

TC39 Iterator Helpers

Search...

TABLE OF CONTENTS

3.1.3 Iterator.prototype

The *Iterator prototype object*:

- is *%Iterator.prototype%*.
- has a [[Prototype]] internal slot whose value is %Object.prototype%.
- is an ordinary object.

3.1.3.1 Iterator.prototype.constructor

`Iterator.prototype.constructor` is an accessor property with attributes { [[Enumerable]]: **false**, [[Configurable]]: **true** }. The [[Get]] and [[Set]] attributes are defined as follows:

3.1.3.1.1 get Iterator.prototype.constructor

The value of the [[Get]] attribute is a built-in function that requires no arguments. It performs the following steps when called:

1. Return %Iterator%.

3.1.3.1.2 set Iterator.prototype.constructor

The value of the [[Set]] attribute is a built-in function that takes an argument *v*. It performs the following steps when called:

1. Perform ? SetterThatIgnoresPrototypeProperties(**this** value, %Iterator.prototype%, **"constructor"**, *v*).
2. Return **undefined**.

NOTE Unlike the "constructor" property on most built-in prototypes, for web-compatibility reasons

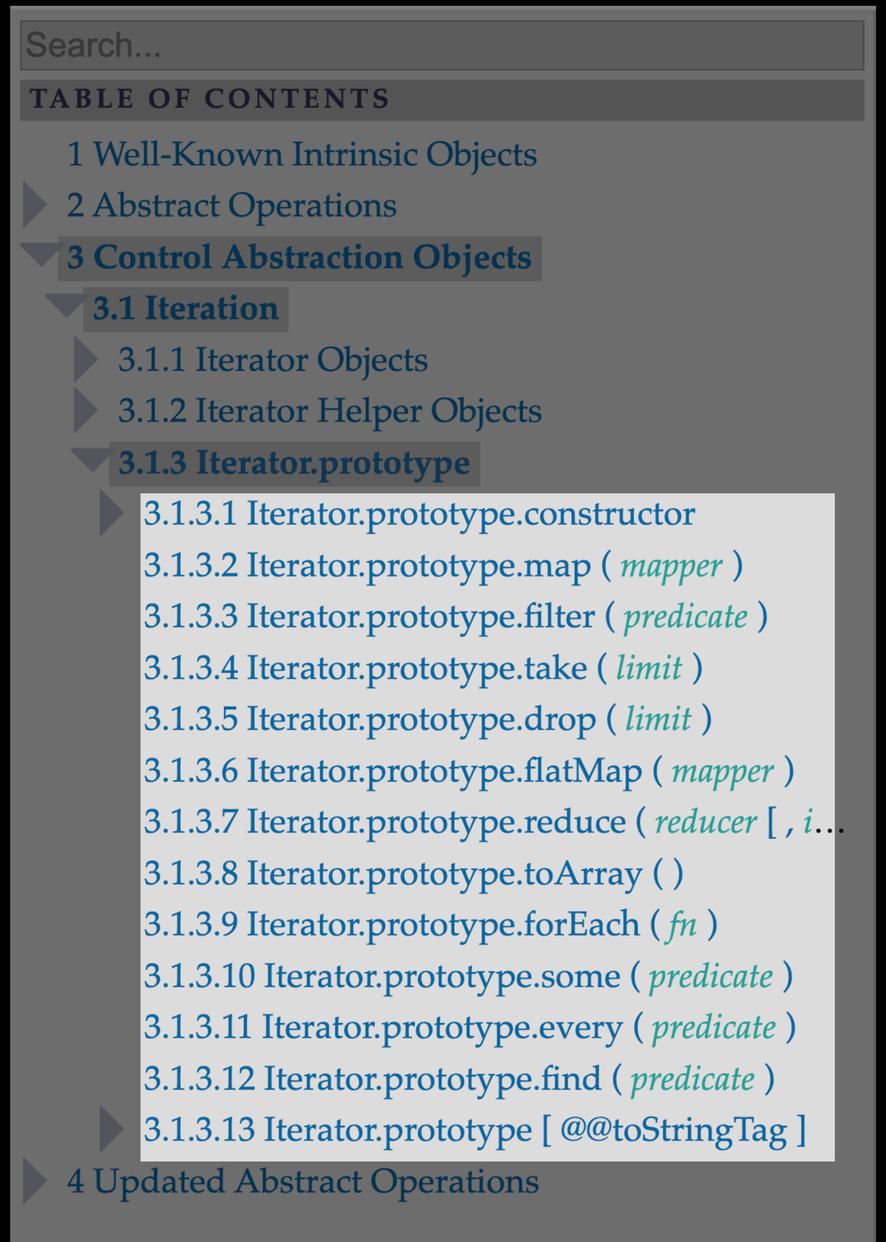
TC39 Iterator Helpers

Search...

TABLE OF CONTENTS

@domfarolino

TC39 Iterator Helpers



@domfarolino

Operators

```
const x = [1, 2, 3];
x.map(v => v * 2);
// [2, 4, 6]
```


dom@chromium.org

Operators

```
const x = [1, 2, 3];
x.filter(v => v % 2 !== 0);
// [1, 3]
```


dom@chromium.org

Operators

```
x = [1, 2, 3];

x.flatMap(n => {
  return [n, n + .1, n + .2];
});

// [1, 1.1, 1.2, 2, 2.1, 2.2, 3, 3.1, 3.2]
```


dom@chromium.org

Operators

```
x = [1, 2, 3];

x.flatMap(n => {
  return [n, n + .1, n + .2];
});

// [1, 1.1, 1.2, 2, 2.1, 2.2, 3, 3.1, 3.2]
```


dom@chromium.org

Operators

```
x = [1, 2, 3];

x.flatMap(n => {
  return [n, n + .1, n + .2];
});

// [1, 1.1, 1.2, 2, 2.1, 2.2, 3, 3.1, 3.2]
```


dom@chromium.org

Operators

```
x = [1, 2, 3];

x.flatMap(n => {
  return [n, n + .1, n + .2];
});

// [1, 1.1, 1.2, 2, 2.1, 2.2, 3, 3.1, 3.2]
```


dom@chromium.org

Operators

```
x = [1, 2, 3];

x.flatMap(n => {
  return [n, n + .1, n + .2];
});

// [1, 1.1, 1.2, 2, 2.1, 2.2, 3, 3.1, 3.2]
```


dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {









  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[ ]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {









  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

1

[ ]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {









  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

1

[ ]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

1

[ ]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

1

[ ]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[ ]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[ ]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

1

[ ]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

1

[1,

]

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[1,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

2

[1,

]

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

2

[1,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[1,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[1, *]*

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

1.1

[1,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(
    complete: () => console.log('done!'),
  });
```

1.1

[1, 1.1,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[1, 1.1,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

3

Observable

[1, 1.1, 1.2]

[1, 1.1,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[1, 1.1,]

dom@chromium.org

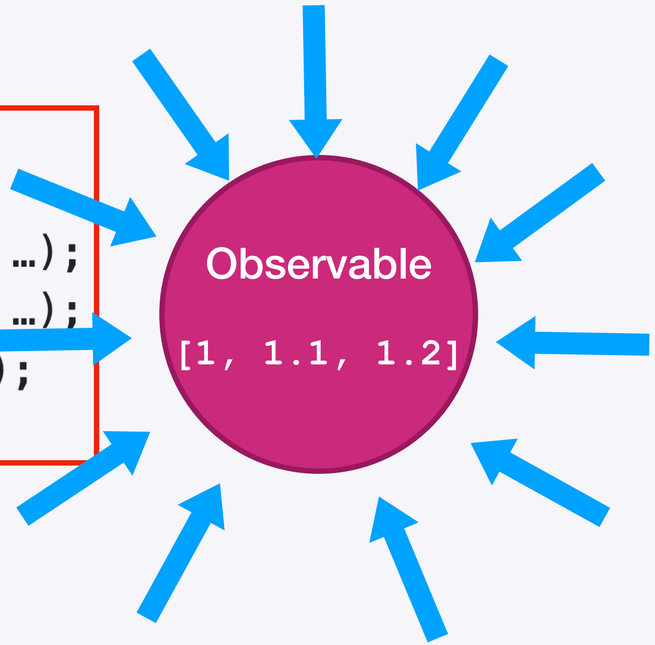
```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

3


```
[ 1, 1.1,                    ]
```


dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```




```
[ 1, 1.1,                    ]
```


dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```


```
[ 1, 1.1,                    ]
```

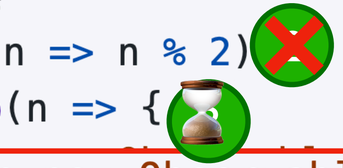

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```


```
[ 1, 1.1,                    ]
```


dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```



[1, 1.1,]

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

1.2

[1, 1.1, 1.2,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```


```
[1, 1.1, 1.2,                ]
```


dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

3

```
[1, 1.1, 1.2,                ]
```

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```


```
[1, 1.1, 1.2,                ]
```


dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```


```
[ 1, 1.1, 1.2,                ]
```

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```


```
[1, 1.1, 1.2,                ]
```

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

3

[1, 1.1, 1.2, 3,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

3.1

[1, 1.1, 1.2, 3,]

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

3.1

[1, 1.1, 1.2, 3, 3.1,]

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

Observable

[3, 3.1, 3.2]

3.2

[1, 1.1, 1.2, 3, 3.1,]

dom@chromium.org

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

3.2

[1, 1.1, 1.2, 3, 3.1, 3.2]

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[1, 1.1, 1.2, 3, 3.1, 3.2]

```
const observable = …;

observable
  .filter(n => n % 2)
  .flatMap(n => {
    return new Observable(subscriber => {
      setTimeout(() => subscriber.next(n), …);
      setTimeout(() => subscriber.next(n + .1), …);
      setTimeout(() => subscriber.next(n + .2), …);
      setTimeout(() => subscriber.complete(), …);
    });
  })
  .take(6)
  .subscribe({
    next: n => console.log(n),
    complete: () => console.log('done!'),
  });
```

[1, 1.1, 1.2, 3, 3.1, 3.2]

```
[Exposed=*]
interface Observable

  // Observable-returning operators.
  //
  Observable takeUntil(any notifier);
  Observable map(Mapper mapper);
  Observable filter(Predicate predicate);
  Observable take(unsigned long long amount);
  Observable drop(unsigned long long amount);
  Observable flatMap(Mapper mapper);
  Observable switchMap(Mapper mapper);
  Observable finally(VoidFunction callback);

  // Promise-returning operators.
  Promise<sequence<any>> toArray(optional SubscribeOptions options = {});
  Promise<undefined> forEach(Visitor callback, optional SubscribeOptions options = {});
  Promise<boolean> every(Predicate predicate, optional SubscribeOptions options = {});
  Promise<any> first(optional SubscribeOptions options = {});
  Promise<any> last(optional SubscribeOptions options = {});
  Promise<any> find(Predicate predicate, optional SubscribeOptions options = {});
  Promise<boolean> some(Predicate predicate, optional SubscribeOptions options = {});
  Promise<any> reduce(Reducer reducer, optional any initialValue, optional SubscribeOption
};
```


dom@chromium.org

```
[Exposed=*]
interface Observable

  // Observable-returning operators.
  //
  Observable takeUntil(any notifier);
  Observable map(Mapper mapper);
  Observable filter(Predicate predicate);
  Observable take(unsigned long long amount);
  Observable drop(unsigned long long amount);
  Observable flatMap(Mapper mapper);
  Observable switchMap(Mapper mapper);
  Observable finally(VoidFunction callback);

  // Promise-returning operators.
  Promise<sequence<any>> toArray(optional SubscribeOptions options = {});
  Promise<undefined> forEach(Visitor callback, optional SubscribeOptions options = {});
  Promise<boolean> every(Predicate predicate, optional SubscribeOptions options = {});
  Promise<any> first(optional SubscribeOptions options = {});
  Promise<any> last(optional SubscribeOptions options = {});
  Promise<any> find(Predicate predicate, optional SubscribeOptions options = {});
  Promise<boolean> some(Predicate predicate, optional SubscribeOptions options = {});
  Promise<any> reduce(Reducer reducer, optional any initialValue, optional SubscribeOption
};
```


dom@chromium.org

Unsubscription & teardown

dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  setInterval(() => subscriber.next(i++), 1000);



});
```


dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  setInterval(() => subscriber.next(i++), 1000);



});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  }



}                        );
```


dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  setInterval(() => subscriber.next(i++), 1000);



});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  }
```

```
}                          );
```


dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  setInterval(() => subscriber.next(i++), 1000);



});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  }
    💥 💥
}                          );
```


dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  setInterval(() => subscriber.next(i++), 1000);



});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  }
    💥 💥 💥
}                          );
```


dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  setInterval(() => subscriber.next(i++), 1000);



});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  }
    💥 💥 💥 💥
}                          );
```


dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);



});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  }
    💥 💥 💥 💥
}                          );
```


dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);

  subscriber.addTeardown(() => clearInterval(token));
});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  }
    💥 💥 💥 💥
}                          );
```


dom@chromium.org

```
const ac = new AbortController();
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);

  subscriber.addTeardown(() => clearInterval(token));
});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  }
    💥 💥 💥 💥
}                          );
```


dom@chromium.org

```
const ac = new AbortController();
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);

  subscriber.addTeardown(() => clearInterval(token));
});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  } else {
    ac.abort();
  }
}                       );
```


dom@chromium.org

```
const ac = new AbortController();
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);

  subscriber.addTeardown(() => clearInterval(token));
});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  } else {
    ac.abort();
  }
}  {signal: ac.signal});
```

```
const ac = new AbortController();
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);

  subscriber.addTeardown(() => clearInterval(token));
});

source.subscribe(v => {
  if (v <= 100) {
    console.log(v);
  } else {
    ac.abort();
  }
}, {signal: ac.signal});
```

Terminating operators

`take()`, `takeUntil()`, `first()`, ...

dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);
  subscriber.addTeardown(() => clearInterval(token));
});
```


dom@chromium.org

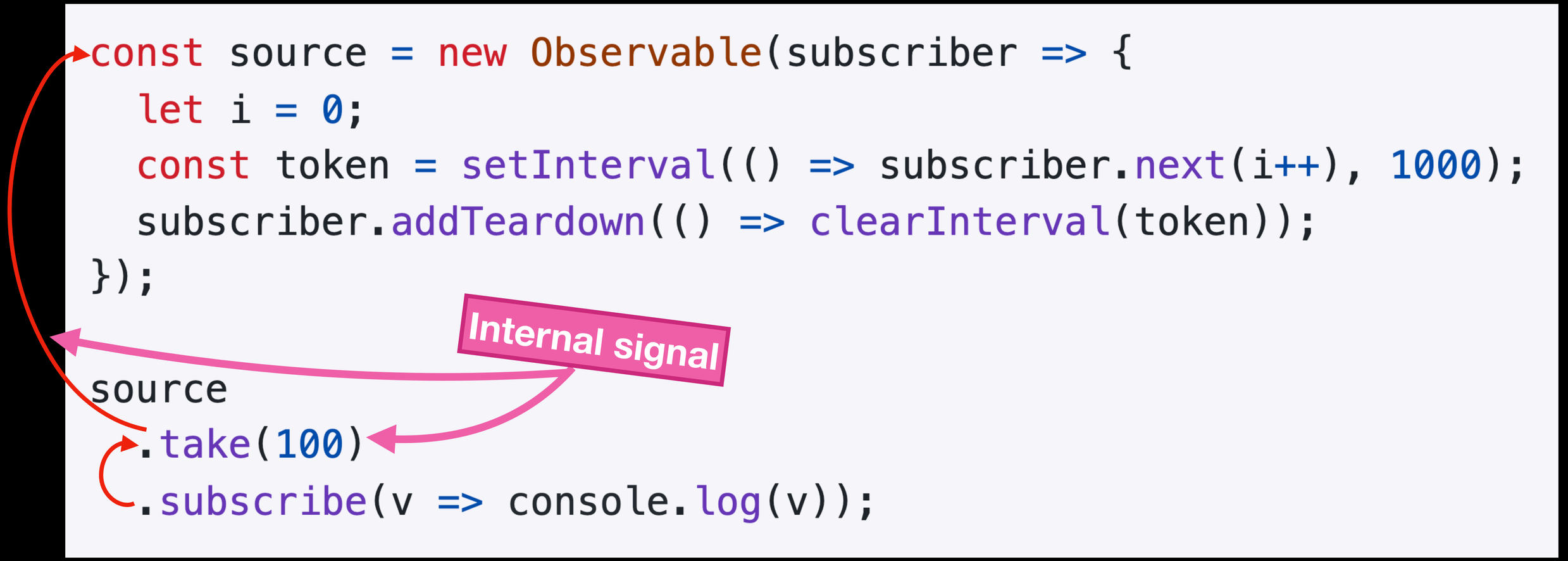
```
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);
  subscriber.addTeardown(() => clearInterval(token));
});

source
  .take(100)
  .subscribe(v => console.log(v));
```


dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);
  subscriber.addTeardown(() => clearInterval(token));
});

source
  .take(100)
  .subscribe(v => console.log(v));
```

```
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);
  subscriber.addTeardown(() => clearInterval(token));
});

source
  .take(100)
  .subscribe(v => console.log(v));
```


dom@chromium.org



dom@chromium.org

```
const source = new Observable(subscriber => {
  let i = 0;
  const token = setInterval(() => subscriber.next(i++), 1000);
  subscriber.addTeardown(() => clearInterval(token));
});

source
  .take(100)
  .subscribe(v => console.log(v));
```


dom@chromium.org

observable

Public

Edit Pins | Unwatch 30 | Fork 12 | Star 520

master | 4 Branches | 0 Tags | Go to file t | Add file | Code

benlesh and domfarolino Adds the catch operator to the README (#87) 98a4f07 · 2 days ago 94 Commits

.github/workflows	Add write permissions for spec-prod	8 months ago
.gitignore	Add empty specification (#77)	8 months ago
.pr-preview.json	Add empty specification (#77)	8 months ago
.prettierrc	Format markdown file with prettier (#63)	7 months ago
CONTRIBUTING.md	Adding baseline CONTRIBUTING.md	9 months ago
LICENSE.md	Adding baseline LICENSE.md	9 months ago
README.md	Adds the catch operator to the README (#87)	2 days ago
security-privacy-questionnaire.md	Correct security & privacy answer	9 months ago
spec.bs	Spec the `last()` operator (#144)	last month
w3c.json	Adding baseline w3c.json	9 months ago

README | License

Observable

About

Observable API proposal

wicg.github.io/observable/

javascript html dom observable web-application reactive-programming whatwg observables

Readme
View license
Activity
Custom properties
520 stars
30 watching
12 forks
Report repository

Releases

No releases published
Create a new release

Packages

No packages published

WICG / observable

Type / to search

ode Issues 38 Pull requests 3 Discussions Actions Projects Security Insights Settings

observable Public

Edit Pins Unwatch 30 Fork 12 Star 520

master 4 Branches 0 Tags Go to file t Add file Code

benlesh and domfarolino Adds the catch operat 2 days ago 94 Commits

.github/workflows		8 months ago
.gitignore		8 months ago
.pr-preview.json		8 months ago
.prettierrc		7 months ago
CONTRIBUTING.md		9 months ago
LICENSE.md		9 months ago
README.md		2 days ago
security-privacy-questionnaire.md	Correct security & privacy answer	9 months ago
spec.bs		last month
w3c.json	Adding baseline w3c.json	9 months ago

github.com/WICG/observable

README License

Observable

About

Observable API proposal

wicg.github.io/observable/

javascript html dom observable web-application reactive-programming whatwg observables

Readme
View license
Activity
Custom properties
520 stars
30 watching
12 forks
Report repository

Releases

No releases published
Create a new release

Packages

No packages published

chrome://flags/#observable-api

github.com/WICG/observable

@domfarolino