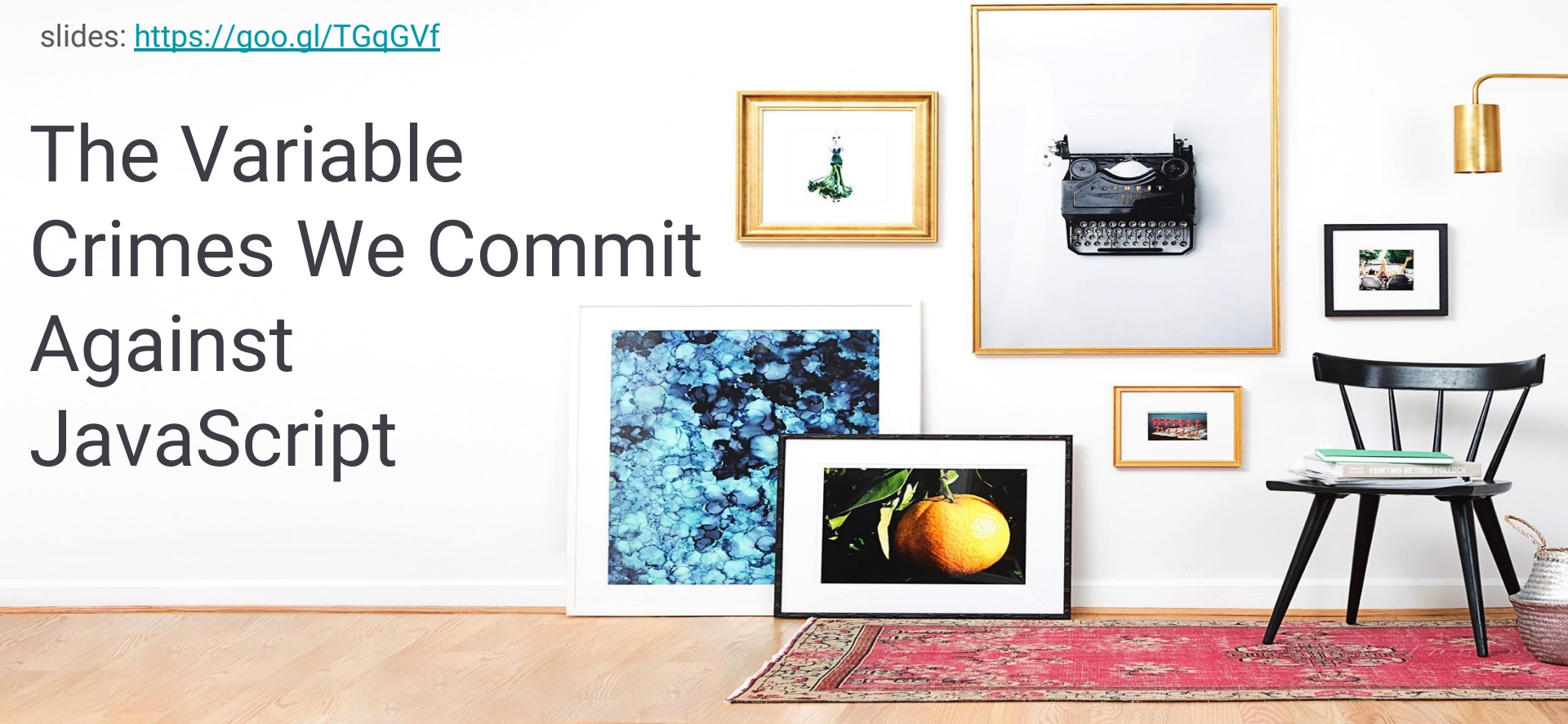
FRAMEBRIDGE

code: https://talks.jgrodel.com/201805BluegrassDataScienceGroup/examples.js

slides: https://goo.gl/TGqGVf

The Variable Crimes We Commit Against JavaScript



Speaker

Julka Grodel

@jgrodel

julka

julka_loves_pink

Framebridge

Principal Software Engineer

@framebridge

FRAMEBRIDGE

@jgrodel

use code BluegrassData for 15% off first order thru 06/30/2018

scope
types of functions
hoisting
closure
arrays & objects
es6

scope


```
pink = 1;
console.log(pink);
console.log(window.pink);
```

```
pink = 1;
console.log(pink);
console.log(window.pink);
```

```
pink = 1;
console.log(pink); // 1
console.log(window.pink);
```

```
pink = 1;
console.log(pink); // 1
console.log(window.pink);
```

```
pink = 1;
console.log(pink); // 1
console.log(window.pink); // 1
```


```
var fuschia = 2;
console.log(fuschia);
console.log(window.fuschia);
```

```
var fuschia = 2;
console.log(fuschia);
console.log(window.fuschia);
```

```
var fuschia = 2;
console.log(fuschia); // 2
console.log(window.fuschia);
```

```
var fuschia = 2;
console.log(fuschia); // 2
console.log(window.fuschia);
```

```
var fuschia = 2;
console.log(fuschia); // 2
console.log(window.fuschia); // 2
```

immediately invoked anonymous functions

function(){...}

function(){...}()

(function(){...}())

(function(){...}());


```
(function() {
   magenta = 3;
   console.log(magenta);
   console.log(window.magenta);
}());
console.log(magenta);
console.log(window.magenta);
```

```
(function() {
   magenta = 3;
   console.log(magenta);
   console.log(window.magenta);
}());
console.log(magenta);
console.log(window.magenta);
```

```
(function() {
   magenta = 3;
   console.log(magenta); // 3
   console.log(window.magenta);
}());
console.log(magenta);
console.log(window.magenta);
```

```
(function() {
   magenta = 3;
   console.log(magenta); // 3
   console.log(window.magenta);
}());
console.log(magenta);
console.log(window.magenta);
```

```
(function() {
   magenta = 3;
   console.log(magenta); // 3
   console.log(window.magenta); // 3
}());
console.log(magenta);
console.log(window.magenta);
```

```
(function() {
   magenta = 3;
   console.log(magenta); // 3
   console.log(window.magenta); // 3
}());
console.log(magenta);
console.log(window.magenta);
```

```
(function() {
   magenta = 3;
   console.log(magenta); // 3
   console.log(window.magenta); // 3
}());
console.log(magenta); // 3
console.log(window.magenta);
```

```
(function() {
   magenta = 3;
   console.log(magenta); // 3
   console.log(window.magenta); // 3
}());
console.log(magenta); // 3
console.log(window.magenta);
```

```
(function() {
   magenta = 3;
   console.log(magenta); // 3
   console.log(window.magenta); // 3
}());
console.log(magenta); // 3
console.log(window.magenta); // 3
```


```
(function() {
   var rouge = 4;
   console.log(rouge);
   console.log(window.rouge);
}());
console.log(window.rouge);
console.log(rouge);
```

```
(function() {
   var rouge = 4;
   console.log(rouge);
   console.log(window.rouge);
}());
console.log(window.rouge);
console.log(rouge);
```

```
(function() {
   var rouge = 4;
   console.log(rouge);
   console.log(window.rouge);
}());
console.log(window.rouge);
console.log(rouge);
```

```
(function() {
   var rouge = 4;
   console.log(rouge); // 4
   console.log(window.rouge);
}());
console.log(window.rouge);
console.log(rouge);
```

```
(function() {
   var rouge = 4;
   console.log(rouge); // 4
   console.log(window.rouge);
}());
console.log(window.rouge);
console.log(rouge);
```

```
(function() {
   var rouge = 4;
   console.log(rouge); // 4
   console.log(window.rouge); // undefined
}());
console.log(window.rouge);
console.log(rouge);
```

```
(function() {
   var rouge = 4;
   console.log(rouge); // 4
   console.log(window.rouge); // undefined
}());
console.log(window.rouge);
console.log(rouge);
```

```
(function() {
   var rouge = 4;
   console.log(rouge); // 4
   console.log(window.rouge); // undefined
}());
console.log(window.rouge); // undefined
console.log(rouge);
```

```
(function() {
   var rouge = 4;
   console.log(rouge); // 4
   console.log(window.rouge); // undefined
}());
console.log(window.rouge); // undefined
console.log(rouge);
```

```
(function() {
   var rouge = 4;
   console.log(rouge); // 4
   console.log(window.rouge); // undefined
}());
console.log(window.rouge); // undefined
console.log(rouge); // Uncaught ReferenceError: rouge is not defined
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush);
    blush = 6;
    console.log(blush);
    var rose = 7;
    console.log(rose);
  }());
  console.log(blush);
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush);
    blush = 6;
    console.log(blush);
    var rose = 7;
    console.log(rose);
  }());
  console.log(blush);
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush);
    var rose = 7;
    console.log(rose);
  }());
  console.log(blush);
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush);
    var rose = 7;
    console.log(rose);
  }());
  console.log(blush);
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose);
  }());
  console.log(blush);
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose);
  }());
  console.log(blush);
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose); // 7
  }());
  console.log(blush);
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose); // 7
  }());
  console.log(blush);
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose); // 7
  }());
  console.log(blush); // 6
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose); // 7
  }());
  console.log(blush); // 6
  console.log(rose);
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose); // 7
  }());
  console.log(blush); // 6
  console.log(rose); // Uncaught ReferenceError: blush is not defined
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose); // 7
  }());
  console.log(blush); // 6
  console.log(rose); // Uncaught ReferenceError: blush is not defined
}());
console.log(blush);
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose); // 7
  }());
  console.log(blush); // 6
  console.log(rose); // Uncaught ReferenceError: blush is not defined
}());
console.log(blush); // Uncaught ReferenceError: blush is not defined
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose); // 7
  }());
  console.log(blush); // 6
  console.log(rose); // Uncaught ReferenceError: blush is not defined
}());
console.log(blush); // Uncaught ReferenceError: blush is not defined
console.log(rose);
```

```
(function() {
  var blush = 5;
  (function() {
    console.log(blush); // 5
    blush = 6;
    console.log(blush); // 6
    var rose = 7;
    console.log(rose); // 7
  }());
  console.log(blush); // 6
  console.log(rose); // Uncaught ReferenceError: blush is not defined
}());
console.log(blush); // Uncaught ReferenceError: blush is not defined
console.log(rose); // Uncaught ReferenceError: rose is not defined
```


```
function() {
   function() {
      function() {
         function() {
            //...
         }();
      }();
   }();
}();
```

```
function() {
  function() {
    function() {
      function() {
        //...
      }();
    }();
  }();
}();
```

```
function() {
  function() {
    function() {
      function() {
        //...
      }();
    }();
  }();
}();
```

```
function() {
  function() {
    function() {
      function() {
        //...
      }();
    }();
  }();
}();
```

```
function() {
   function() {
      function() {
         function() {
            //...
         }();
      }();
   }();
}();
```

```
function() {
  function() {
    function() {
      function() {
        //...
      }();
    }();
  }();
}();
```


```
if (true) {
    var maroon = 29;
    console.log(maroon);
}
console.log(maroon);
```

```
if (true) {
    var maroon = 29;
    console.log(maroon);
}
console.log(maroon);
```

```
if (true) {
   var maroon = 29;
   console.log(maroon);  // 29
}
console.log(maroon);
```

```
if (true) {
   var maroon = 29;
   console.log(maroon);  // 29
}
console.log(maroon);
```

```
if (true) {
   var maroon = 29;
   console.log(maroon);  // 29
}
console.log(maroon);  // 29
```


```
var terracotta = 19;
function blaze(){
   console.log(terracotta);
   var terracotta = 20;
};
blaze();
```

```
var terracotta = 19;
function blaze(){
   console.log(terracotta);
   var terracotta = 20;
};
blaze();
```

hoisting

```
var terracotta = 19;
function blaze(){
   console.log(terracotta);
   var terracotta = 20;
};
blaze();
```

```
var terracotta = 19;
function blaze(){
   console.log(terracotta);
   var terracotta;
   terracotta = 20;
};
blaze();
```

```
var terracotta = 19;
function blaze(){
   var terracotta;
   console.log(terracotta);
   terracotta = 20;
};
blaze();
```

```
var terracotta = 19;
function blaze(){
   var terracotta;
   console.log(terracotta);
   terracotta = 20;
};
blaze();
```

```
var terracotta = 19;
function blaze(){
   var terracotta;
   console.log(terracotta); // undefined
   terracotta = 20;
};
blaze();
```

named functions

```
function pear() {
    // ...
};
```

anonymous functions

```
var pear = function() {
    // ...
};
```


```
chocolate();

function chocolate(){
   console.log("yes please!");
};
```

```
chocolate();

function chocolate(){
   console.log("yes please!");
};
```

```
chocolate();

function chocolate(){
   console.log("yes please!");
};
```

```
chocolate();

function chocolate(){
   console.log("yes please!");
};
```

```
chocolate(); // yes please!

function chocolate(){
   console.log("yes please!");
};
```

```
var chocolate;

chocolate = function(){
   console.log("yes please!");
};

chocolate();
```

```
var chocolate;

chocolate = function(){
   console.log("yes please!");
};

chocolate();
```

```
var chocolate;

chocolate = function(){
   console.log("yes please!");
};

chocolate();
```

```
var chocolate;

chocolate = function(){
   console.log("yes please!");
};

chocolate();
```

closure


```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
  var carrot = 0;
  var greaterThanTwo = function() {
    carrot++;
    return carrot > 2;
  };
  return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
  var carrot = 0;
  var greaterThanTwo = function() {
    carrot++;
    return carrot > 2;
  };
  return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
  var carrot = 0;
  var greaterThanTwo = function() {
    carrot++;
    return carrot > 2;
  };
  return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
  var carrot = 0;
  var greaterThanTwo = function() {
    carrot++;
    return carrot > 2;
  };
  return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
  var carrot = 0;
  var greaterThanTwo = function() {
    carrot++;
    return carrot > 2;
  };
  return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon());
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon()); // false
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon()); // false
console.log(salmon());
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon()); // false
console.log(salmon()); // false
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon()); // false
console.log(salmon()); // false
console.log(salmon());
```

```
var marmalade = function() {
   var carrot = 0;
   var greaterThanTwo = function() {
      carrot++;
      return carrot > 2;
   };
   return greaterThanTwo;
};
var salmon = marmalade();
// don't have direct access to carrot
console.log(salmon()); // false
console.log(salmon()); // false
console.log(salmon()); // true
```

arrays & objects


```
console.log(typeof 'yellow');
```

```
console.log(typeof 'yellow'); // string
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7);
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7); // number
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7); // number
console.log(typeof true);
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7); // number
console.log(typeof true); // boolean
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7); // number
console.log(typeof true); // boolean
console.log(typeof {});
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7); // number
console.log(typeof true); // boolean
console.log(typeof {}); // object
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7); // number
console.log(typeof true); // boolean
console.log(typeof {}); // object
console.log(typeof []);
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7); // number
console.log(typeof true); // boolean
console.log(typeof {}); // object
console.log(typeof []); // object
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7); // number
console.log(typeof true); // boolean
console.log(typeof {}); // object
console.log(typeof []); // object
console.log(Array.isArray([]));
```

```
console.log(typeof 'yellow'); // string
console.log(typeof 7); // number
console.log(typeof true); // boolean
console.log(typeof {}); // object
console.log(typeof []); // object
console.log(Array.isArray([])); // true
```


```
var canary = {type: 'bird', flies: true};
var bumblebee = canary;
bumblebee.type = 'insect';
console.log(canary.type);
```

```
var canary = {type: 'bird', flies: true};
var bumblebee = canary;
bumblebee.type = 'insect';
console.log(canary.type);
```

```
var canary = {type: 'bird', flies: true};
var bumblebee = canary;
bumblebee.type = 'insect';
console.log(canary.type); // insect
```

ES6
let & const


```
let green = 21;
green = 22;
```


```
const sage = 23;
```

```
const sage = 23;
sage = 24;
```

```
const sage = 23;
sage = 24; // Uncaught TypeError: Assignment to constant variable.
```


```
const lime = [];
lime.push(25);
console.log(lime);
lime = [26];
```

```
const lime = [];
lime.push(25);
console.log(lime);
lime = [26];
```

```
const lime = [];
lime.push(25);
console.log(lime);
lime = [26];
```

```
const lime = [];
lime.push(25);
console.log(lime);
lime = [26];
```

```
const lime = [];
lime.push(25);
console.log(lime); // [25]
lime = [26];
```

```
const lime = [];
lime.push(25);
console.log(lime); // [25]
lime = [26];
```

```
const lime = [];
lime.push(25);
console.log(lime); // [25]
lime = [26]; // Uncaught TypeError: Assignment to constant variable.
```

ES6
code blocks


```
let olive = 28;
if (true) {
   let olive = 29;
   console.log(olive);
}
console.log(olive);
```

```
let olive = 28;
if (true) {
    let olive = 29;
    console.log(olive);
}
console.log(olive);
```

```
let olive = 28;
if (true) {
   let olive = 29;
   console.log(olive); // 29
}
console.log(olive);
```

```
let olive = 28;
if (true) {
   let olive = 29;
   console.log(olive); // 29
}
console.log(olive);
```

```
let olive = 28;
if (true) {
   let olive = 29;
   console.log(olive); // 29
}
console.log(olive);  // 28
```


```
let moss = 36;
{
   let moss = 37;
   console.log(moss);
}
console.log(moss);
```

```
let moss = 36;
{
   let moss = 37;
   console.log(moss);
}
console.log(moss);
```

```
let moss = 36;
{
   let moss = 37;
   console.log(moss);  // 37
}
console.log(moss);
```

```
let moss = 36;
{
   let moss = 37;
   console.log(moss);  // 37
}
console.log(moss);
```

```
let moss = 36;
{
   let moss = 37;
   console.log(moss);  // 37
}
console.log(moss);  // 36
```


```
for (let emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald);
}
console.log(emerald);
```

```
for (let emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald);
}
console.log(emerald);
```

```
for (let emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald);
}
console.log(emerald);
```

```
for (let emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald); // 30, 31, 32, 33, 34
}
console.log(emerald);
```

```
for (let emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald); // 30, 31, 32, 33, 34
}
console.log(emerald);
```

```
for (let emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald); // 30, 31, 32, 33, 34
}
console.log(emerald);
                          // Uncaught ReferenceError: emerald is not defined
```


```
for (var emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald);
}
console.log(emerald);
```

```
for (var emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald);
}
console.log(emerald);
```

```
for (var emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald);
}
console.log(emerald);
```

```
for (var emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald); // 30, 31, 32, 33, 34
}
console.log(emerald);
```

```
for (var emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald); // 30, 31, 32, 33, 34
}
console.log(emerald);
```

```
for (var emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald); // 30, 31, 32, 33, 34
}
console.log(emerald); // 35
```

```
let emerald;
for (emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald); // 30, 31, 32, 33, 34
}
console.log(emerald);
```

```
let emerald;
for (emerald = 30; emerald <= 34; emerald++) {
   console.log(emerald); // 30, 31, 32, 33, 34
}
console.log(emerald); // 35
```

scope
types of functions
hoisting
closure
arrays & objects
es6

Thank you!

code: https://talks.jgrodel.com/201805BluegrassDataScienceGroup/examples.js

slides: https://goo.gl/TGqGVf

use code BluegrassData for 15% off first order thru 06/30/2018