|  |  |
| --- | --- |
| var a = 1;  var b = 2;  const a = 1;  const b = 2; | Good? |
| let count = 1;  if (true) {  count += 1;  }  var count = 1;  if (true) {  count += 1;  } |  |
| {  let a = 1;  const b = 1;  }  console.log(a); | True? |
| const item = {};  const item = new Object(); | Bad? |
| function getKey(k) {  return `a key named ${k}`;  }  const obj = {  id: 5,  name: 'San Francisco',  [getKey('enabled')]: true,  };  const obj = {  id: 5,  name: 'San Francisco',  };  obj[getKey('enabled')] = true; | Bad? |
| const atom = {  value: 1,  addValue(value) {  return atom.value + value;  },  };  const atom = {  value: 1,  addValue: function (value) {  return atom.value + value;  },  }; | Good? |
| const lukeSkywalker = 'Luke Skywalker';  const obj = {  lukeSkywalker: lukeSkywalker,  };  const obj = {  lukeSkywalker,  }; | Bad? |
| const anakinSkywalker = 'Anakin Skywalker';  const lukeSkywalker = 'Luke Skywalker';  const obj = {  lukeSkywalker,  anakinSkywalker,  episodeOne: 1,  twoJediWalkIntoACantina: 2,  episodeThree: 3,  mayTheFourth: 4,  };  const obj = {  episodeOne: 1,  twoJediWalkIntoACantina: 2,  lukeSkywalker,  episodeThree: 3,  mayTheFourth: 4,  anakinSkywalker,  }; | Good? |
| const bad = {  'foo': 3,  'bar': 4,  'data-blah': 5,  };  const good = {  foo: 3,  bar: 4,  'data-blah': 5,  }; | Good? |
| console.log(Object.prototype.hasOwnProperty.call(object, key)  import has from 'has';  console.log(has.call(object, key)); | Best? |
| const someStack = [];  someStack[someStack.length] = 'abracadabra';  someStack.push('abracadabra'); | Bad? |
| const items = [1, 2, 3, 4, 6];  A)  const itemsCopy = [...items];  B)  const len = items.length;  const itemsCopy = [];  let i;  for (i = 0; i < len; i += 1) {  itemsCopy[i] = items[i];  } | Good? |
| const foo = [1, 2, 3, 4];  const bar = [5, 6, 7];  A)  const baz = Array.from(foo, bar);  B)  const baz = […foo, …bar] | Bad?  Best? |
| const name = 'Capt. Janeway';  const name = "Capt. Janeway"; | Bad? |
| function sayHi(name) {  return `How are you, ${name}?`;  }  function sayHi(name) {  return ‘How are you, ${name}?’;  } | Good? |
| const errorMessage = 'This is a super long error that was thrown because \  of Batman. When you stop to think about how Batman had anything to do \  with this, you would get nowhere \  fast.';  const errorMessage = 'This is a super long error that was thrown because ' +  'of Batman. When you stop to think about how Batman had anything to do ' +  'with this, you would get nowhere fast.';  const errorMessage = 'This is a super long error that was thrown because of Batman. When you stop to think about how Batman had anything to do with this, you would get nowhere fast.'; | Bad? |
| function sayHi(name) {  return 'How are you, ' + name + '?';  }  function sayHi(name) {  return `How are you, ${name}?`;  } | Good? |
| const foo = '\'this\' is "quoted"';  const foo = '\'this\' \i\s \"quoted\"'; | Good? |
| function handleThings(opts = {}, name) {  // ...  }  function handleThings(name, opts = {}) {  // ...  } | Good? |
| var add = new Function('a', 'b', 'return a + b');  var subtract = Function('a', 'b', 'return a - b'); | Good? |
| function f3(a) {  const b = a || 1;  // ...  }  function f2(a) {  if (!a) { a = 1; }  // ...  }  function f1(a) {  a = 1;  // ...  } | Good? |
| const x = [1, 2, 3, 4, 5];  console.log(...x); | Bad? |
| const x = [1, 2, 3, 4, 5];  A)  console.log(...x);  B)  x.map(function (i) {  console.log(i);  }); | Bad? |
| const t = [1, 2, 3, 4, 5];  A)  t.map((x) => {  const y = x + 1;  return x \* y;  });  B)  t.map(function (x) {  const y = x + 1;  return x \* y;  }); | Good? |
| const itemHeight = item => (item.height > 256 ? item.largeSize : item.smallSize);  const itemHeight = item => item.height > 256 ? item.largeSize : item.smallSize; | Good? |