Ecma/TC39/2014/036

Should We Eliminate Early Errors on parameter/let declaration name conflicts?

Allen Wirfs-Brock

Legacy

```
function (x) {
  var x;
  console.log(x);
)(42);  //output 42

function (x) {
  function x(){};
  console.log(x);
)(42);  //output “function x(){}”

function (x) {
  function x(){};
  var x;
  console.log(x);
)(42);  //output “function x(){}”
```

ES6 (uncontroversial)

```
function (x) {
  var x;
  let x;   //early error
);

function (x) {
  function x(){};
  let x;  //early error
);

function (x) {
  function x(){};
  var x;
  let x;   //early error
);
```

Current ES6 Spec (controversy)

```
function (x) {
  var x;
  let x;   //early error
);

function (x) {
  let x;  //early error
);

try {
} catch (x) {
  let x;  //early error
}
```

What do other C syntax languages do:

- C++, §3.3.3: A parameter name shall not be redeclared in the outermost block of the function definition nor in the outermost block of any handler associated with a function-try-block.
- Java, §6.4: It is a compile-time error if the name of a formal parameter is redeclared as a local variable of the method or constructor; ...
- C#, §10.6: A method declaration creates a separate declaration space for parameters, type parameters and local variables. Names are introduced into this declaration space by ... the formal parameter list of the method and by local variable declarations in the *block* of the method. It is an error for two members of a method declaration space to have the same name.