# AST

Abstract syntax tree (AST), or just syntax tree, is a tree representation of the abstract syntactic structure of source code written in a programming language.

# Terms

- Formal
- Actual
- Attribute
- Dispatch
    - Static
    - Dynamic

# Example of AST (#1)

- Given the following expression:

```
1 + 2
```

- The output AST would be:

```
+ BinaryExpression
 - type: +
 - left_value:
  LiteralExpr:
   value: 1
 - right_value:
  LiteralExpr:
   value: 2
```

# Example of AST (#2)

- Given the following expression:

```
if (2 > 6) {
    var d  = 90
    console.log(d)
}
```

# Example of AST (#2) Cont.

The output AST would be:

```
IfStatement
 - condition
  + BinaryExpression
   - type: >
   - left_value: 2
   - right_value: 6
 - body
  [
    - Assign
        - left: 'd';
        - right:
            LiteralExpr:
            - value: 90
    - MethodCall:
        - instanceName: console
        - methodName: log
        - args: [
        ]
  ]
```

# Lab exercise

Let's try to construct the AST for the following cool program

```
{
  var i: Int = 0;
  var b: Boolean = false;
  while (i < 20) {
    if (b) {
      out("Pow!")
    } else ();

    b = !b;
    i = i + 1
  }
}
```

# Solution

```
@265 = block:22
    @264 = let:22 'i 'Int
      @240 = int_lit:12 '0
      @263 = block:22
          @262 = let:22 'b 'Boolean
            @241 = bool_lit:13 false
            @261 = block:22
                @260 = loop:22
                  @244 = lt:14
                    @242 = variable:14 'i
                    @243 = int_lit:14 '20
                  @259 = block:21
                      @251 = cond:17
                        @245 = variable:15 'b
                        @249 = block:17
                            @248 = dispatch:17 'out
                              @247 = variable:17 'this
                                @246 = string_lit:16 'Pow!
                        @250 = unit:17
                      @254 = assign:19 'b
                        @253 = comp:19
                          @252 = variable:19 'b
                      @258 = assign:21 'i
                        @257 = add:21
                          @255 = variable:20 'i
                          @256 = int_lit:21 '1
```

# Lab assignment

Follow "Cool Tour" to construct "Queens.cool" given it's AST. Basically reverse engineer the cool program from its AST.