1 Number type: double | Literal: Number, null, boolean, object {}, array, string, regex, template, func

JSLint - Doug Crakford | Access object properties w/ dot/array | Reflection: code that inspects structure of code

function name (args) {body}; name optional | Closure foundation for encapsulation

"let" = block scope, "var" hoists | Inherit from super (parent) class w/ extends | • floating point inexact

Shallow copy = properties shared
Deep copy = properties NOT shared

array.slice = SC
array.splice = DC

Awful • scope • parseInt
• global variables • + operator adds/concat
• F by values true when compared

Regex options:
• g - global, i - insensitive, m - multiline

function expression = const foo = function() {}

Regex cheat sheet:
^ = match beginning of string
$ = match end of string
\ = escape character
() = capture group
. = match any single character
* = look for 0 or more of whatever is left of *
+ = 1 or more
? = optional character
[a-z] = any char in range

STRICT MODE:
• Eliminates silent errors
• Throws exceptions
• Fixes mistakes that make optz difficult
• Prohibits possible future syntax

MCR STRICT:
• Can't create globals
• Assigning NaN throws error
• Prevents duplicate properties in object
• Prevents deleting things you shouldn't
• Forbids octal syntax

Cohesion vs coupling:
Things within module should relate (high cohesion). Dependencies between modules should be minimized (low coupling)

spread operator (...) can spread array (...array) into comma seperated list

Spread works on objects, makes shallow copy

Prototype inheritance creates link
When asking for property x.p, looks for p in x → p in prototype of x → till chain ends

Kyler Ashby - SCRATCH WORK

```
let areaCircle = 0;
let areaTriangle = 0;

let Area = (function() {
    let radius = getElementById("c_radius").value();
    if (areaCircle !== 0;) && (radius === undefined) &&
        getElementById("answer").innerHTML = areaCircle;
```

Kyler Ashby - SCRATCH WORK

```
let areaCircle = 0;
let areaTriangle = 0;

let Area = (function() {

    return {
        circle: function() {
            let radius = getElementById("c_radius").value();
            if ((areaCircle !== 0) && (radius === undefined)) {
                getElementById("answer").innerHTML = areaCircle;
            } else {
                areaCircle = toString(3.14 * (radius ^ 2));
            };
        },
        triangle: function() {
            let base = getElementById("t_base").value();
            let height = getElementById("t_height").value();
            if ((areaTriangle !== 0) && ((base === undefined) || (height ===
            undefined))) {
                getElementById("answer").innerHTML = areaTriangle;
            } else {
                areaTriangle = toString(0.5 * base * height);
            };
        }

)};
```

Kyler Ashby - SCRATCH WORK

```
let Area = (function(){
  let valueCircle = 0;
  let valueTriangle = 0;
    return {
      circle: function(){
        let radius = getElementById("c_radius").value();
        valueCircle = toString(3.14*(radius^2));
        getElementById("answer").innerHTML = valueCircle;
      },
      triangle: function(){
        let base = getElementById("t_base").value();
        let height = getElementById("t_height").value();
        valueTriangle = toString(0.5*base*height);
        getElementById("answer").innerHTML = valueTriangle;
      }
})();
```

Kyler Ashby - FINAL ANSWER

```
let areaCircle = 0;
let areaTriangle = 0;

let Area = (function() {
  return {
    circle: function() {
      let radius = getElementById("c_radius").value();
      if ((areaCircle !== 0) && ((radius === undefined)) {
        getElementById("answer").innerHTML = areaCircle;
      } else if ((areaCircle === 0) && (radius === undefined)) {
        getElementById("answer").innerHTML = "Input radius";

      } else {
        areaCircle = toString(3.14 * (radius ^ 2));
        getElementById("answer").innerHTML = areaCircle;
      };
    },
    triangle: function () {
      let base = getElementById("t_base").value();
      let height = getElementById("t_height").value();
      if ((areaTriangle !== 0) && ((base === undefined) || height === undefined))){
        getElementById("answer") innerHTML = areaTriangle;
    } else if ( ((areaTriangle === 0) && ((base ===undefined) || height ===
    undefined))) {
        getElementById("answer").innerHTML = "Input base and/or
        height";
      } else {
        areaTriangle = toString(0.5 * base * height);
        getElementById("answer").innerHTML = areaTriangle;
      };
    }
});
```