JAVASCRIPT CHEAT SHEET

- Programming language
- Strict syntax requirements
- Interpreted; object-oriented; dynamic typing

- Access web content (elements)
- Modify web content
- React to events
- Retrieve data without page reload (AJAX)

PRIMITIVE DATA TYPES

Type	*Example(s)*	*Falsey Value(s)*
number	`1, 3, 5.6, -284`	`0, -0, NaN`
string	`“hello”, “Mr. F”`	`“”`
boolean	`true, false`	`FALSE`
null	`null`	`null`
undefined	`undefined`	`undefined`
symbol (ES6)	`Symbol(“first”)`	*none*

SOME TYPE CONVERSIONS

`parseFloat(“0.05”)`	*Convert string to floating point number*
`parseInt(“1000”, 10)`	*Convert string to integer number in base 10*
`89.toString()`	*Convert number to string*

OBJECTS

- used to store related data and methods
- key-value pairs
- syntax: `{ key: value, key2: value2 }`
- dot notation: `obj.key = newValue;`
- bracket notation: `obj[‘key’] = newValue;`

```
pet = {
        name: ‘Theo’,
        bark: function(){
                console.log(‘woof!’);
              },
        age: 3
      }
```

ARRAYS

- used to store sequential data
- integer keys (indices) starting from 0
- syntax: `[ el0, el1, el2 ]`
- bracket notation for indices: `arr[0] = 7;`

```
coinFlips = [0, 1, 1, 1, 0];
```

PROGRAMMING CONCEPTS

variable	`var thing = 5;`	*Store some data and give it a name for reuse.*

conditional	`if (size == 'large'){`
	`  price = 4.99;`
	`} else if (size == 'medium'){`
	`  price = 2.00;`
	`} else {`
	`  price = 1.29;`
	`}`
		Decide which line of code to run next based on current conditions.

loop
- *initialization*
- *continue condition*
- *update*
	`var j=10;`
	`while (j>=0;){`
	`  console.log(j);`
	`  j=j-1;`
	`}`
		Repeat an action while a condition is true.

	`for (var k=10; k>=0; k=k-1){`
	`  console.log(k);`
	`}`
		Repeat an action for each value in a sequence.

function	`function add (num1, num2){`
	`  return num1 + num2;`
	`}`
		Function definition: store some behavior and give it a name for reuse.

	`add(6, 9);`	*Function call: run the code now on specific values*

BOOLEAN COMPARISON OPERATORS

`==`	*Equal to*	`>`	*Greater than*
`===`	*Strict equal to*	`<`	*Less than*
`!=`	*Not equal to*	`>=`	*Greater than or equal to*
`!==`	*Strict not equal to*	`<=`	*Less than or equal to*

BOOLEAN LOGICAL OPERATORS

`&&`	*and*
`||`	*or*
`!`	*not*