WDI Fundamentals Unit 3

Expressions & Variables Cheat Sheet

Here are some notes on what's been covered in this chapter; feel free to copy this and extend it to make your own cheatsheet.

Expressions

- An expression is a statement composed of values/data and operators
- Some common data types are Numbers, Strings, and Booleans
- An operator takes in some number of inputs, but outputs/evaluates to a single value.
- To determine how an expression is evaluated, look at what each operator's inputs are and, if necessary, generate an expression tree to illustrate the expression's structure.

Variables

- The purpose of variables is to store and re-use the values created from a computation.
- A variable is assigned a value using the `=` operator. First, the expression to the right of the `=` is evaluated. Then, this value is assigned to the variable to the left of the `=`. Finally, the `=` operator evaluates to the value that has just been assigned.
- To use the value that a variable is storing, simply include the variable in an expression. An expression contaiing variables will evaluate just like one without variables, except that the variables will themselves be evaluated as part of the expression. As before, it is possible to draw an expression tree to illustrate the expression's structure.
- When a variable is redefined, it retains **no knowledge** of any prior values it may have held.
- A variable may be redefined 'in place' using an expression like `x = x + 1` (or its short-hand, `x += 1`).
- An expression like `x = y` only means that the value that `y` had been holding is now also being held in `x`; **it does not imply any lasting relationship between x and y**.

Special Cases

- When a variable is created, but is not assigned a value, it will be evaluated as `null`.

- Any type of value, including `null`, can be passed into a logical operator as an input; based on whether these inputs are either 'truthy' or 'falsey', and what type of operator you're dealing with, the operator will behave in different ways.

Comparison Operators

Operator	Meaning	True expressions
==	Equality	10 == '10'
===	Strict equality	(2 * 5) === 10
!=	Inequality	9 != 10
!==	Strict inequality	'10' !== 10
>	Greater than	20 > 10
>=	Greater than or equal	'10' >= 10
<	Less than	10 < 30
<=	Less than or equal	'10'<= 10

Logical Operators

Logical operators work on Boolean values to produce Boolean results.

AND operator `&&`

Condition 1	Condition 2	Result
true	true	true

Condition 1	Condition 2	Result
true	false	false
false	true	false
false	false	false

OR operator `||`

Condition 1	Condition 2	Result
true	true	true
true	false	true
false	true	true
false	false	false

NOT operator `!`

Condition	Result
true	false
false	true

We can use parentheses to change the order of operations for logical operators, just like we do with mathematical ones.