

1. JavaScript tries to 'understand' what you mean, and may transform the type of a value to get everything to work. With `==` you allow for this transformation to happen, and `false == 0` would return `true`. With `===` you don't allow this transformation and you want to know whether the two values are exactly equal to each other.
2. Variables usually either only live inside a function (local) or outside (and inside too) of a function (global). With a closure, a variable can be the return value of a function, which itself is also a function. This way, the variable is still accessible after a function has 'finished', because it still 'lives on' in its parent function.
3. Functions that either take in other functions in as an argument, or which return another function.
4. Something that selects elements of the DOM and returns the elements (one or all) with the corresponding tag.