# Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ StudentID: \_\_\_\_\_\_\_\_\_\_\_

You can test regular expressions online: <https://regexr.com/> or <http://regex101.com/>

or print a cheat sheet: <https://www.cheatography.com/davechild/cheat-sheets/regular-expressions/>

## Activity 1: Regular Expressions

Regular expressions can be used in input pattern attributes and javascript to detect a string's content and format. Explain what the following regex looks for, in what format, and how it works:

^(\(04\)|04|\+614)[ ]?\d{4}[ ]?\d{4}$

(04) or 01 or +614 , 0-1 space, 4 digits, 0-1 space, 4 digits

Can you write an improved version? One that allows for more flexibility?

^(\(04\)|04|\+614|+61 4)([ ]?\d){8}$

Write a regex that matches characters found in a (Western) name field. Make sure you can match these names?

* <https://en.wikipedia.org/wiki/Gerard_%27t_Hooft>
* <http://www.imdb.com/name/nm0001876>
* <http://www.imdb.com/name/nm0000375>

"^[A-Za-z\-'., ]{1,100}$"

|  |
| --- |
| Once these tests are complete, use the working regexes in your assignment. |

## Activity 2: Javascript Code Generation

What type of HTML element is the following javascript trying to create? Draw a picture below the code box.

|  |
| --- |
| 1. <p>Birth Year: 2. <select name='dob'> 3. <script> 4. var year = new Date().getFullYear(); 5. for ( var i=year; i<=(year-50); i-- ) 6. document.write('<option value="'+i+'">'+i+'</option>'); 7. </script> 8. </select> 9. </p> |

Which line is the "deliberate mistake" on? Fix the error so that the code will run correctly.

Line 5: no “var”,

Missing {}

Option value “i”

Think of a practical use for the corrected code, and rewrite line 5 so that no one alive is discriminated against.

i=year, i<=(year-150), i--

How might you adapt the "practical use" code so that only those that are 18 and older some time this year qualify?

## Activity 3: Javascript Arrays: Indexed and Associative

An array is a sequential list. The simplest is a **numerically indexed array**, **indexed** by **integers** starting at **0.** To initialise a simple 1D indexed array in javascript with one statement, we can use the following syntax:

|  |
| --- |
| var numbers = [ 'zero', 'one', 'two', 'three', 'four' ]; |

What does numbers[2] return?

‘two’

Another way of looking at this array is that each index is a key pointing to a value(eg in the above example, the 2 in numbers[2] is a key pointing to a value). To initialise a simple 1D **associative array** in javascript with one statement, we can use the following syntax:

|  |
| --- |
| var letters = { zero:'A', one:'B', two:'C', three:'D', four:'E' }; |

What does letters['two'] return? What about letters.two?

* Null
* ‘C’

A programmer will usually want to iterate throughout the whole array. The for-in loop simplifies the process of iterating through an array. Compare the standard for loop syntax with the for-in loop syntax:

|  |
| --- |
| // Using a for loop  for ( var key=0; key<numbers.length; key++ ) {  document.write( key + ': ' + numbers[key] + "<br>" );  }  // Using a for-in loop  for ( var key in letters ) {  document.write( key + ': ' + letters[key] + "<br>" );  } |

## 

## Activity 4: Web Storage: localStorage and sessionStorage

localStorage and sessionStorage are two new HTML5 data structures (ie variables) that store strings into associative arrays on the client's machine. Data takes the form of keys and values in both structures.

**Handy Tip:** Before using localStorage and sessionStorage make sure that the browser supports it by performing a typeof check. The example below shows localStorage, but the syntax is the same for sessionStorage.

|  |
| --- |
| if (typeof(Storage) !== "undefined") {    // There are three ways to set a storage variable  localStorage.setItem('key','value'); // or ...  localStorage['key'] = 'value'; // or ..  localStorage.key = 'value';  // Many ways NOT to set a storage variable  localStorage.setItem('key', 5); // fails, 5 is not a string  localStorage.setItem('key', true); // fails, true is not a string  localStorage.setItem('key', [1,2,3]); // fails, an array is not a string  // and one way to unset a storage variable  localStorage.removeItem('key');  } |

What is the difference between localStorage and sessionStorage?

LocalStorage doesn’t expire while sessionStorage will be cleared after thesession ends

Why is using web storage better than using cookies, also stored client side?

Web storage can store much more data (5MB) compared to cookies (4KB)