Week 4 - Strings 1 - Wednesday Recap

This week's challenges focus on *strings*. Some or all of these might be useful:

.length *.split* *.join* *.concat* *for...in loops* *switch/case*

Parens Valid

Create a function that, given an input string, returns a boolean whether parentheses in that string are valid.

```
// Track nest level: always >= 0; end as 0
function parensValid(str)
{
  if (typeof str !== "string")
  { return false; }            // fast-fail

  var numParens = 0;
  for (var idx=0; idx<str.length; idx++) {
    if (str[idx] == "(") { numParens++;  }
    if (str[idx] == ")") { numParens--;  }
    if (numParens < 0)   { return false; }
  }
  return (numParens === 0);
}
```

Braces Valid

Return whether various parentheses, braces and brackets in the given string are valid.

```
// Any close must match most recent open!
function bracesValid(str)
{
  if (typeof str !== "string")
  { return false; }            // fast-fail

  var matches =
          { ")": "(", "}": "{", "]": "[" };
  var braceArr = [];
  for (var idx = 0;idx < str.length;idx++){
    switch (str[idx]) {
      case "(":
      case "{":
      case "[":
        braceArr.push(str[idx]);
        break;
      case ")":
      case "}":
      case "]":
        if (  braceArr[braceArr.length - 1]
           != matches[str[idx]] ) {
          return false;
        }
        braceArr.pop();
        break;
      default:
        continue;
    }
  }
  return (braceArr.length === 0);
}
```

Week 4 - Strings 1 - Thursday

This week's challenges focus on *strings*. Some or all of these might be useful:

.length *.split* *.join* *.concat* *for...in loops* *switch/case*

Is Palindrome

Strings like "Able was I, ere I saw Elba" or "Madam, I'm Adam" could be considered *palindromes*, because (if we ignore spaces, punctuation and capitalization) the letters are the same from front and back.

Create a function that returns a boolean whether the string is a *strict* palindrome. For "a x a" or "racecar", return true. Do **not** ignore spaces, punctuation and capitalization: if given "Dud" or "oho!", return false.

Answer:

Longest Palindrome

For this challenge, we will look not only at the entire string, but also substrings within it.

For a string, return the longest palindromic substring. Given "what up, dada?", return "dad". Given "not much", return "n". Include spaces as well: given "My favorite racecar erupted!", return "e racecar e".

Answer:

Tomorrow: books that all end the same….