

String Processing

String Processing

Jerry Cain and Eric Roberts
CS 106J
May 3, 2017

String Calisthenics

Let's review some **String** methods you learned last time:

- ✓ "AEIOUaeiou".length
- ✓ "ABCDEFGF".charAt(6)
- ✓ "Harry Potter".indexOf("a")
- ✓ "Harry Potter".indexOf("a", 6)
- ✓ "Harry Potter".lastIndexOf("tt")
- ✓ "bumfuzzle".substring(3, 7)
- ✓ "cabotage".substring(1, 1)

Generating Acronyms

- An *acronym* is a word formed by taking the first letter of each word in a sequence, as in
 - "North American Free Trade Agreement" → "NAFTA"
 - "not in my back yard" → "nimby"
 - "self-contained underwater breathing apparatus" → "scuba"
- The text describes and implements two versions of a function `acronym(str)` that generates an acronym for `str`:
 - The first version searches for spaces in the string and includes the following character in the acronym. This version, however, fails for acronyms like *scuba*, in which some of the words are separated by hyphens rather than spaces.
 - The second version looks at every character and keeps track of whether the algorithm is scanning a word formed composed of sequential letters. This version correctly handles *scuba* as well as strings that have leading, trailing, or multiple spaces.

First Version of the `acronym` Function

```
acronym("Graphics Interchange Format")
function acronym(str) {
  var result = str.charAt(0);
  var sp = str.indexOf(" ");
  while ( sp !== -1 ) {
    result += str.charAt(sp + 1);
    sp = str.indexOf(" ", sp + 1);
  }
  return result;
}
```

```
Console
> acronym("Graphics Interchange Format")
GIF
>
```

PPAcronym

```
acronym("Laughing Out Loud")
function acronym(str) {
  var result = "";
  var inWord = false;
  for ( var i = 0 ; i < str.length ; i++ ) {
    var ch = str.charAt(i);
    if ( isLetter(ch) ) {
      if (!inWord) result += ch;
      inWord = true;
    } else {
      inWord = false;
    }
  }
  return result;
}
```

```
Console
> acronym("Laughing Out Loud")
LOL
```

Translating Pig Latin to English

Section 6.4 works through the design and implementation of a program to convert a sentence from English to Pig Latin. In this dialect, the Pig Latin version of a word is formed by applying the following rules:

1. If the word begins with a consonant, the `wordToPigLatin` function moves the initial consonant string to the end of the word and then adds the suffix *ay*, as follows:
scram → *amscray*
2. If the word begins with a vowel, `wordToPigLatin` generates the Pig Latin version simply by adding the suffix *way*, like this:
apple → *appleway*
3. If the word contains no vowels at all, `wordToPigLatin` returns the original word unchanged.

Pseudocode for the Pig Latin Program

```
function toPigLatin(str) {  
  Initialize a variable called result to hold the growing string.  
  for (each character position in str) {  
    if (the current character is a letter) {  
      if (we're not yet scanning a word) Remember the start of this word.  
    } else {  
      if (we're scanning a word) {  
        Call wordToPigLatin to translate the word.  
        Append the translated word to the result variable.  
      }  
    }  
  }  
  if (we're still scanning a word) {  
    Call wordToPigLatin and append the translated word to result  
  }  
}  
  
function wordToPigLatin(word) {  
  Find the first vowel in the word.  
  If there are no vowels, return the original word unchanged.  
  If the vowel appears in the first position, return the word concatenated with "way".  
  Divide the string into two parts (head and tail) before the vowel.  
  Return the result of concatenating the tail, the head, and the string "ay".  
}
```

Simulating the PigLatin Program

```
toPigLatin("this is pig latin")  
function toPigLatin(str) {  
  function wordToPigLatin(word) {  
    var vp = findFirstVowel(word);  
    if (vp === -1) {  
      return word;  
    } else if (vp === 0) {  
      return word + "way";  
    } else {  
      var head = word.substring(0, vp);  
      var tail = word.substring(vp);  
      return tail + head + "ay";  
    }  
  }  
  
  word vp head tail  
  
  Console  
  > toPigLatin("this is pig latin")  
  isthay isway igpay atinlay
```