In order to implement the Hangman game we need to be able to take a secret word, and a single character guess, and use these to update a string that records which letters have been guessed:

<table>
<thead>
<tr>
<th>secret, guess, record</th>
<th>new record</th>
</tr>
</thead>
<tbody>
<tr>
<td>“banana”, ‘e’ , “_____”</td>
<td>“_____”</td>
</tr>
<tr>
<td>“banana”, ‘a’ , “_____”</td>
<td>“_a_a_a”</td>
</tr>
<tr>
<td>“banana”, ‘n’ , “_a_a_a”</td>
<td>“_anana”</td>
</tr>
</tbody>
</table>
Assume the players successful guesses are stored in a string called `playerword`. We can check if we have guessed the character at a position in this word using `charAt`

If `playerword` has value “a_a_a”,
then `playerword.charAt(3)` has value ‘a’.

We can use an index variable to step through the `playerword` in order to see if any/all of them have been guessed correctly.
Motivating Arrays (III)

\[ i = 0; \text{found} = \text{False} \]

\[ i < \text{playerword length} \]

Yes

\[ \text{ith char} = \_ \]

Yes

\[ \text{found} = \text{True} \]

No

\[ i = i + 1 \]
We have a variable `guess` holding the most recent player guess. We can test if the $i$th character of player word has/has not been guessed:

\[
\text{playerword.charAt}(i) == "_"
\]

Example, if we have found “a” at position 1 ($i == 1$) so we want to change

"______"

to

"_a_____"

We need to use the following code:

```
playerword = playerword.substring(0,i-1) + guess + playerword.substring(i+1,playerword.length-1);
```
About Arrays (I)

If we declare

    Array(5)

then we have 5 things, numbered 0 ... 4!
Instead using strings and having to write:

    playerword.charAt(i)

we introduce an array:

    var playerword = new Array ('_','_',...,'_');

(Here we need to know and fix the secret word length in advance)
Now we can easily access the \textit{i\textsuperscript{th}} position

    playerword[i]
A better approach — get the secret word into variable `secretword` using `prompt`. Then setup the `playerword` array as follows:

```java
for( i=0; i < secretword.length; i++ ) {
    playerword[i] = '_'
}
```
We have a variable `guess` holding the most recent player guess. We can test if the $i$th character of player word has/has not been guessed:

```python
playerword[i] == "_"
```

Example, if we have found “a” at position 1 ($i == 1$) so we want to change

```
"______"
```

to

```
"_a____"
```

We simply use the following code:

```python
playerword[i] = guess
```