Text Processing
Chris Piech and Mehran Sahami
CS106A, Stanford University
Housekeeping

• Happy Friday
• Mehran made an extra example video!
• Chris AMA tomorrow at 9am. Come ask me things!
  – https://us.edstem.org/courses/325/discussion/59604
• Diagnostic review sessions
• Mid quarter evaluations
Fake Medicine was a Problem

700,000 deaths a year from fake malaria and tuberculosis drugs [1]

Chris' Favorite Program

Bright Simons

Piech and Sahami, CS106A, Stanford University
Underlying Puzzle

Counterfeiter

User

You (Distributor)
You (Distributor)

Counterfeiter

User

Piech and Sahami, CS106A, Stanford University
Revisit this problem...
Learning Goals

1. Write string algorithms that loops over each character
Review
Review of lists

```python
def main():
    my_list = [415, 650, 999]
    print(my_list)
```

```mermaid
diagram FlowChart
  my_list --> 415 - 650 - 999
  0 - 1 - 2
```
End Review
Text Problem: Decryption
The spirit is willing but the flesh is weak.

(Russian)

The vodka is good but the meat is rotten.

*This result cost billions of dollars (adjusted for inflation)*
Text Problem: DNA Analysis
How is text represented?
Text is stored using the variable type `string`. A `string` is a sequence of characters.

```python
def main():
    text = "hello!"
    print(text)
```
Hello!
Hello!
def main():
    text = "hello!"

How it is actually stored

Length: 6
def main():
    text = "hello!"

text[index]
All characters in a string have an index.

You can access a character in the sequence via its \textit{index}. 
String Functions

- The `len(string)` function returns the number of characters in the string. This is one larger than the last valid index in the string.
- the `string[i]` function returns the character at a given index.
A string is indexed just like a list! Slices work too.

It is *almost* like it is a list of characters.
Chris@doto Desktop % python3
Python 3.8.1 (v3.8.1:1b293b6006, Dec 18 2019, 14:08:53)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 'Hi mom'
>>> x[0]
'H'
>>> x[1]
'i'
>>> x[2]
' '.
>>> x[3]
'm'
>>> x[4]
'o'
>>> x[5]
'm'
>>> x[6]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: string index out of range
>>>
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```python
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
```

<table>
<thead>
<tr>
<th>example</th>
<th>length</th>
<th>first</th>
</tr>
</thead>
<tbody>
<tr>
<td>H i m o m</td>
<td>6</td>
<td>'H'</td>
</tr>
</tbody>
</table>
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)

Console
6
H

<table>
<thead>
<tr>
<th>example</th>
<th>length</th>
<th>first</th>
</tr>
</thead>
<tbody>
<tr>
<td>H i m o m</td>
<td>6</td>
<td>'H'</td>
</tr>
</tbody>
</table>

Piech and Sahami, CS106A, Stanford University
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    # for loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example))[0, 1, 2, 3, 4, 5]
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length)  # prints 6

    # example of getCharAt
    first = example[0]
    print(first)  # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
def main():
    example = "Hi mom"

    # example of length function
    length = len(example)
    print(length) # prints 6

    # example of getCharAt
    first = example[0]
    print(first) # prints 'H'

    # loop that prints letters one-by-one
    for i in range(len(example)):
        ch = example[i]
        print(ch)
Again there are two for loops

```python
for i in range(len(example)):
    ch = example[i]
    print(i, ch)

for ch in example:
    print(ch)
```
Functions you can call on strings

must know:

<table>
<thead>
<tr>
<th>Function</th>
<th>Code</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>split</td>
<td><code>x.split(' ')</code></td>
<td><code>['this', 'is', 'a', 'Test']</code></td>
</tr>
<tr>
<td>upper</td>
<td><code>x.upper()</code></td>
<td>'#THIS IS A TEST '</td>
</tr>
<tr>
<td>lower</td>
<td><code>x.lower()</code></td>
<td>'#this is a test '</td>
</tr>
<tr>
<td>replace</td>
<td><code>x.replace('is','lol')</code></td>
<td>'#thlol lol a Test'</td>
</tr>
<tr>
<td>find</td>
<td><code>x.find('is')</code></td>
<td>#2</td>
</tr>
<tr>
<td>strip</td>
<td><code>strip()</code></td>
<td>'#this is a test'</td>
</tr>
</tbody>
</table>

good to know:

<table>
<thead>
<tr>
<th>Function</th>
<th>Code</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>startswith</td>
<td><code>x.startswith('th')</code></td>
<td># True</td>
</tr>
<tr>
<td>endswith</td>
<td><code>x.endswith('end')</code></td>
<td># False</td>
</tr>
<tr>
<td>title</td>
<td><code>x.title()</code></td>
<td>'#This Is A Test '</td>
</tr>
<tr>
<td>isalpha</td>
<td><code>x.isalpha()</code></td>
<td>#False</td>
</tr>
<tr>
<td>isdigit</td>
<td><code>'521'.isdigit()</code></td>
<td>#True</td>
</tr>
<tr>
<td>isspace</td>
<td><code>'   '.isspace()</code></td>
<td>#True</td>
</tr>
</tbody>
</table>
Just the number please
DNA  ->  mRNA
Advanced version

How are characters represented?
Some examples:

- `letter_A` = `'A'`
- `plus` = `'+'`
- `zero` = `'0'`
- `space` = `' '`
- `korean_ch` = `'보'`
- `new_line` = `'\n'`  # special
- `tab` = `'\t'`  # special
- `backslash` = `'\\'`  # special
- `backslash` = `'\'`  # special
- `emoji` = `'😃'`
- `first` = `text[0]`
Advanced course

Chars are just a giant enumeration!
The letter A, for example, has the ASCII value 65

<table>
<thead>
<tr>
<th>Code</th>
<th>Char</th>
<th>Code</th>
<th>Char</th>
<th>Code</th>
<th>Char</th>
<th>Code</th>
<th>Char</th>
<th>Code</th>
<th>Char</th>
<th>Code</th>
<th>Char</th>
<th>Code</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>[space]</td>
<td>48</td>
<td>@</td>
<td>64</td>
<td>P</td>
<td>80</td>
<td>`</td>
<td>96</td>
<td>`</td>
<td>112</td>
<td>p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>!</td>
<td>49</td>
<td>1</td>
<td>65</td>
<td>A</td>
<td>81</td>
<td>Q</td>
<td>97</td>
<td>a</td>
<td>113</td>
<td>q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>&quot;</td>
<td>50</td>
<td>2</td>
<td>66</td>
<td>B</td>
<td>82</td>
<td>R</td>
<td>98</td>
<td>b</td>
<td>114</td>
<td>r</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>#</td>
<td>51</td>
<td>3</td>
<td>67</td>
<td>C</td>
<td>83</td>
<td>S</td>
<td>99</td>
<td>c</td>
<td>115</td>
<td>s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>$</td>
<td>52</td>
<td>4</td>
<td>68</td>
<td>D</td>
<td>84</td>
<td>T</td>
<td>100</td>
<td>d</td>
<td>116</td>
<td>t</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>%</td>
<td>53</td>
<td>5</td>
<td>69</td>
<td>E</td>
<td>85</td>
<td>U</td>
<td>101</td>
<td>e</td>
<td>117</td>
<td>u</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>&amp;</td>
<td>54</td>
<td>6</td>
<td>70</td>
<td>F</td>
<td>86</td>
<td>V</td>
<td>102</td>
<td>f</td>
<td>118</td>
<td>v</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>'</td>
<td>55</td>
<td>7</td>
<td>71</td>
<td>G</td>
<td>87</td>
<td>W</td>
<td>103</td>
<td>g</td>
<td>119</td>
<td>w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>(</td>
<td>56</td>
<td>8</td>
<td>72</td>
<td>H</td>
<td>88</td>
<td>X</td>
<td>104</td>
<td>h</td>
<td>120</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>)</td>
<td>57</td>
<td>9</td>
<td>73</td>
<td>I</td>
<td>89</td>
<td>Y</td>
<td>105</td>
<td>i</td>
<td>121</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>*</td>
<td>58</td>
<td>;</td>
<td>74</td>
<td>J</td>
<td>90</td>
<td>Z</td>
<td>106</td>
<td>j</td>
<td>122</td>
<td>z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>+</td>
<td>59</td>
<td>:</td>
<td>75</td>
<td>K</td>
<td>91</td>
<td>[</td>
<td>107</td>
<td>k</td>
<td>123</td>
<td>{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>,</td>
<td>60</td>
<td>&lt;</td>
<td>76</td>
<td>L</td>
<td>92</td>
<td>]</td>
<td>108</td>
<td>l</td>
<td>124</td>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>-</td>
<td>61</td>
<td>=</td>
<td>77</td>
<td>M</td>
<td>93</td>
<td>\</td>
<td>109</td>
<td>m</td>
<td>125</td>
<td>~</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>.</td>
<td>62</td>
<td>&gt;</td>
<td>78</td>
<td>N</td>
<td>94</td>
<td>^</td>
<td>110</td>
<td>n</td>
<td>126</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>/</td>
<td>63</td>
<td>?</td>
<td>79</td>
<td>O</td>
<td>95</td>
<td>111</td>
<td>o</td>
<td>127</td>
<td>[backspace]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* This is only the first half of the table

Using portions of slides by Eric Roberts
Unicode (bigger ASCII)

```
<table>
<thead>
<tr>
<th>320</th>
<th>321</th>
<th>322</th>
<th>323</th>
<th>324</th>
<th>325</th>
<th>326</th>
<th>327</th>
<th>328</th>
<th>329</th>
<th>32A</th>
<th>32B</th>
<th>32C</th>
<th>32D</th>
<th>32E</th>
<th>32F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>330</th>
<th>331</th>
<th>332</th>
<th>333</th>
<th>334</th>
<th>335</th>
<th>336</th>
<th>337</th>
<th>338</th>
<th>339</th>
<th>33A</th>
<th>33B</th>
<th>33C</th>
<th>33D</th>
<th>33E</th>
<th>33F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>340</th>
<th>341</th>
<th>342</th>
<th>343</th>
<th>344</th>
<th>345</th>
<th>346</th>
<th>347</th>
<th>348</th>
<th>349</th>
<th>34A</th>
<th>34B</th>
<th>34C</th>
<th>34D</th>
<th>34E</th>
<th>34F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>350</th>
<th>351</th>
<th>352</th>
<th>353</th>
<th>354</th>
<th>355</th>
<th>356</th>
<th>357</th>
<th>358</th>
<th>359</th>
<th>35A</th>
<th>35B</th>
<th>35C</th>
<th>35D</th>
<th>35E</th>
<th>35F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>360</th>
<th>361</th>
<th>362</th>
<th>363</th>
<th>364</th>
<th>365</th>
<th>366</th>
<th>367</th>
<th>368</th>
<th>369</th>
<th>36A</th>
<th>36B</th>
<th>36C</th>
<th>36D</th>
<th>36E</th>
<th>36F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>370</th>
<th>371</th>
<th>372</th>
<th>373</th>
<th>374</th>
<th>375</th>
<th>376</th>
<th>377</th>
<th>378</th>
<th>379</th>
<th>37A</th>
<th>37B</th>
<th>37C</th>
<th>37D</th>
<th>37E</th>
<th>37F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>380</th>
<th>381</th>
<th>382</th>
<th>383</th>
<th>384</th>
<th>385</th>
<th>386</th>
<th>387</th>
<th>388</th>
<th>389</th>
<th>38A</th>
<th>38B</th>
<th>38C</th>
<th>38D</th>
<th>38E</th>
<th>38F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>3A0</th>
<th>3A1</th>
<th>3A2</th>
<th>3A3</th>
<th>3A4</th>
<th>3A5</th>
<th>3A6</th>
<th>3A7</th>
<th>3A8</th>
<th>3A9</th>
<th>3AA</th>
<th>3AB</th>
<th>3AC</th>
<th>3AD</th>
<th>3AE</th>
<th>3AF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>3B0</th>
<th>3B1</th>
<th>3B2</th>
<th>3B3</th>
<th>3B4</th>
<th>3B5</th>
<th>3B6</th>
<th>3B7</th>
<th>3B8</th>
<th>3B9</th>
<th>3BA</th>
<th>3BB</th>
<th>3BC</th>
<th>3BD</th>
<th>3BE</th>
<th>3BF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>3C0</th>
<th>3C1</th>
<th>3C2</th>
<th>3C3</th>
<th>3C4</th>
<th>3C5</th>
<th>3C6</th>
<th>3C7</th>
<th>3C8</th>
<th>3C9</th>
<th>3CA</th>
<th>3CB</th>
<th>3CC</th>
<th>3CD</th>
<th>3CE</th>
<th>3CF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>3D0</th>
<th>3D1</th>
<th>3D2</th>
<th>3D3</th>
<th>3D4</th>
<th>3D5</th>
<th>3D6</th>
<th>3D7</th>
<th>3D8</th>
<th>3D9</th>
<th>3DA</th>
<th>3DB</th>
<th>3DC</th>
<th>3DD</th>
<th>3DE</th>
<th>3DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>3E0</th>
<th>3E1</th>
<th>3E2</th>
<th>3E3</th>
<th>3E4</th>
<th>3E5</th>
<th>3E6</th>
<th>3E7</th>
<th>3E8</th>
<th>3E9</th>
<th>3EA</th>
<th>3EB</th>
<th>3EC</th>
<th>3ED</th>
<th>3EE</th>
<th>3EF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>3F0</th>
<th>3F1</th>
<th>3F2</th>
<th>3F3</th>
<th>3F4</th>
<th>3F5</th>
<th>3F6</th>
<th>3F7</th>
<th>3F8</th>
<th>3F9</th>
<th>3FA</th>
<th>3FB</th>
<th>3FC</th>
<th>3FD</th>
<th>3FE</th>
<th>3FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>
```
'A' -> 'Z' are sequential.
'a' -> 'z' are sequential.
'0' -> '9' are sequential.

\texttt{ord(ch)}
Functions which take strings

$$x = 'this \ is \ a \ Test'$$

len
ord
hash
<
==
in

len(x)  # 15
ord('A')  # 65
hash(x)  # 2466759895439727657
'abc' < 'zabc'  # True
x == 'this is a Test'  # True
'his' in x  # True

Won't need these too much in CS106A, but they are super clutch in cryptography and datastructures
Strings have some unique properties
Strings are Immutable

- Python strings are **immutable**: once a string has been created you cannot set characters.
- To change a string:
  - *Create a new string* holding the new value you want it to have via concatenation.
  - Reassigning the String variable (that's allowed).
- **Important consequence**: if you pass a String into a function, you are guaranteed your string won't be changed.
Can survive:
-300F to +300F
Massive radiation
The vacuum of space
Strings are Immutable (Take 1)

\[ x = 'abc' \]

\[ x[1] = 'z' \]

Traceback (most recent call last):
...
TypeError: 'str' object does not support item assignment

\[ x = 'azc' \]
Strings are Immutable (Take 2)

original = 'abba'

Can "reassign"

Can't change
Strings are Immutable (Take 1)

original = 'abba'
original = original + '!'
def main():
    s1 = "CS106"
    s2 = "A"
    s3 = "I got an " + s2 + " in " + s1 + s2

    print(s3)

    I got an A in CS106A
Lists are mutable

Strings are immutable

Immutable is a guarantee that a function won’t be cheeky
Many string algorithms use the "loop and construct" pattern.
Reversing a String

Hello!
Reversing a String

Hello!

!
Reversing a String

Hello!
Reversing a String

Hello!

! o
Reversing a String

Hello!

! o
Reversing a String

Hello!

! o l
Reversing a String

Hello!

! o l
Reversing a String

Hello! → !oll
Reversing a String

Hello!

!o l l
Reversing a String

Hello!

!olle
Reversing a String

Hello!

!olleol!
Reversing a String

Hello!

!olleH
Reversing a String

Hello!
Reversing a String

Hello!

H
Reversing a String

Hello!

eH
Reversing a String

Hello!

Hello!
Reversing a String

Hello!

llleH
Reversing a String

Hello!

olleH
Reversing a String

Hello!

!olleH
def main():
    def reverse_string(str):
        result = ""
        for i in range(len(str)):
            result = str[i] + result
        return result

str = input("Enter a string:")
rev = reverse_string(str)
print(str + "spelled backwards is " + rev)

Using portions of slides by Eric Roberts
reverse_string redux

```python
def reverse_string(str):
    result = ""
    for i in range(len(str)):
        result = str[i] + result
    return result

def reverse_string_v2(str):
    result = ""
    for ch in str:
        result = ch + result
    return result

def reverse_string_v3(str):
    ""
    This uses the slice operator in a special way. With no start, no end and a delta of -1, slice reverses.
    ""
    return str[:::-1]
```

Swiss army knife pattern

Medium bear.

One cool trick that will blow your mind
A palindrome is a string that reads the same forwards and backwards.

For example:
- Abba
- Racecar
- Kayak
- Mr. Owl ate my metal worm.
- Go hang a salami! I'm a lasagna hog.
- Elu par cette crapule
Some test cases

- Let's test our program on some examples:
  - Racecar
  - Kayak
  - Mr. Owl ate my metal worm.
  - Go hang a salami! I'm a lasagna hog.
- Will it work?
More Palindromes

Here are some palindromes in other languages:

- بلح تعلق تحت قلعة حلب (Dates hang underneath a castle in Halab)
- 여보, 안경 안보여 (Honey, I can't see my glasses)
- कड़क (a loud thunderous sound)
- 上海自來水來自海上 (Shanghai tap water originates from "above" the ocean)

The comedian Dmitri Martin also has a routine about palindromes check it out at https://www.youtube.com/watch?v=0hUHDlOazIU
Piech and Sahami, CS106A, Stanford University

**Stress Test**

A man, a plan, a caret, a ban, a myriad, a sum, a lac, a liar, a hoop, a pint, a catalpa, a gas, an oil, a bird, a yell, a vat, a caw, a pax, a wag, a tax, a nay, a ram, a cap, a yam, a gay, a tsar, a wall, a car, a luger, a ward, a bin, a woman, a vassal, a wolf, a tuna, a nit, a pall, a fret, a watt, a bay, a daub, a tan, a cab, a datum, a gall, a hat, a tag, a zap, a say, a jaw, a lay, a wet, a gallop, a tug, a trot, a trap, a tram, a torr, a caper, a top, a tonk, a toll, a ball, a fair, a sax, a minim, a tenor, a bass, a passer, a capital, a rut, an amen, a ted, a cabal, a tang, a sun, an ass, a maw, a sag, a jam, a dam, a sub, a salt, an axon, a sail, an ad, a wadi, a radial, a room, a rood, a rip, a tad, a pariah, a revel, a reel, a reed, a pool, a plug, a pin, a peek, a parabola, a dog, a pat, a cud, a nu, a fan, a pal, a rum, a nod, an eta, a lag, an eel, a batik, a mug, a mot, a nap, a maxim, a mood, a leek, a grub, a gob, a gel, a drab, a citadel, a total, a cedar, a tap, a gag, a rat, a manor, a bar, a gal, a cola, a pap, a yaw, a tab, a raj, a gab, a nag, a pagan, a bag, a jar, a bat, a way, a papa, a local, a gar, a baron, a mat, a rag, a gap, a tar, a decal, a tot, a led, a tic, a bard, a leg, a bog, a burg, a keel, a doom, a mix, a map, an atom, a gum, a kit, a baleen, a gala, a ten, a don, a mural, a pan, a faun, a ducat, a pagoda, a lob, a rap, a keep, a nip, a gulp, a loop, a deer, a leer, a lever, a hair, a pad, a tapir, a door, a moor, an aid, a raid, a wad, an alias, an ox, an atlas, a bus, a madam, a jag, a saw, a mass, an anus, a gnat, a lab, a cadet, an em, a natural, a tip, a caress, a pass, a baronet, a minimax, a sari, a fall, a ballot, a knot, a pot, a rep, a carrot, a mart, a part, a tort, a gut, a poll, a gateway, a law, a jay, a sap, a zag, a tat, a hall, a gamut, a dab, a can, a tabu, a day, a batt, a waterfall, a patina, a nut, a flow, a lass, a van, a mow, a nib, a draw, a regular, a call, a war, a stay, a gam, a yap, a cam, a ray, an ax, a tag, a wax, a paw, a cat, a valley, a drib, a lion, a saga, a plat, a catnip, a pooh, a rail, a calamus, a dairyman, a bater, a canal – Panama!
Remember!

Counterfeiter

User

You (Distributor)
Can you solve it?

Bright Simons

Piech and Sahami, CS106A, Stanford University