

CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

- This lecture will be recorded

Frequently Asked Questions

From email

Frequently Asked Questions

From email

- **I am not sure how to submit the Lab.**

Frequently Asked Questions

From email

- **I am not sure how to submit the Lab.**
You don't submit the lab, you read the lab.

Frequently Asked Questions

From email

- **I am not sure how to submit the Lab.**

*You don't submit the lab, you **read the lab**.*

*When you are done, start working on this week's 10 programming assignments
(this week we will be working on programs 1-10, which is batch 1)*

Frequently Asked Questions

From email

- **I am not sure how to submit the Lab.**

You don't submit the lab, you read the lab.

*When you are done, start working on this week's 10 programming assignments
(this week we will be working on programs 1-10, which is batch 1)*

- **Can I work ahead?**

Frequently Asked Questions

From email

- **I am not sure how to submit the Lab.**

You don't submit the lab, you read the lab.

When you are done, start working on this week's 10 programming assignments (this week we will be working on programs 1-10, which is batch 1)

- **Can I work ahead?**

Absolutely! Submission is open on Gradescope, 3 weeks before the deadline.

Frequently Asked Questions

From email

- **I am not sure how to submit the Lab.**

You don't submit the lab, you read the lab.

When you are done, start working on this week's 10 programming assignments (this week we will be working on programs 1-10, which is batch 1)

- **Can I work ahead?**

Absolutely! Submission is open on Gradescope, 3 weeks before the deadline.

IMPORTANT: Students who work on the due dates in this class tend to miss deadlines and fall behind. If, instead, you work on programs the week of the associated lecture, you will have time to ask for help if you get stuck and still make the deadline.

Frequently Asked Questions

From email

- **I am not sure how to submit the Lab.**

You don't submit the lab, you read the lab.

When you are done, start working on this week's 10 programming assignments (this week we will be working on programs 1-10, which is batch 1)

- **Can I work ahead?**

Absolutely! Submission is open on Gradescope, 3 weeks before the deadline.

IMPORTANT: Students who work on the due dates in this class tend to miss deadlines and fall behind. If, instead, you work on programs the week of the associated lecture, you will have time to ask for help if you get stuck and still make the deadline.

- **When is the midterm?**

Frequently Asked Questions

From email

- **I am not sure how to submit the Lab.**

You don't submit the lab, you read the lab.

When you are done, start working on this week's 10 programming assignments (this week we will be working on programs 1-10, which is batch 1)

- **Can I work ahead?**

Absolutely! Submission is open on Gradescope, 3 weeks before the deadline.

IMPORTANT: Students who work on the due dates in this class tend to miss deadlines and fall behind. If, instead, you work on programs the week of the associated lecture, you will have time to ask for help if you get stuck and still make the deadline.

- **When is the midterm?**

There is no midterm. Instead there's required class quizzes, programming assignments and the final exam.

Today's Topics



- **For-loops**
- `range()`
- Variables
- Characters
- Strings

In Pairs or Triples...

Some review and some novel challenges:

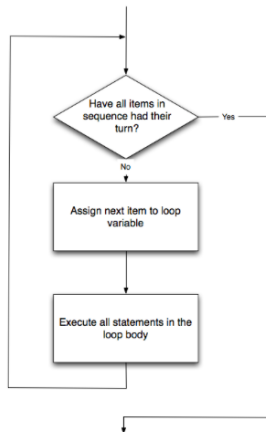
```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

Python Tutor

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

(Demo with pythonTutor)

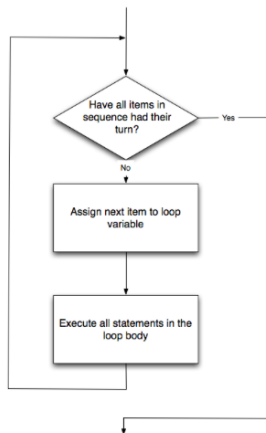
for-loop



```
for i in list:  
    statement1  
    statement2  
    statement3
```

How to Think Like CS, §4.5

for-loop



How to Think Like CS, §4.5

```
for i in list:  
    statement1  
    statement2  
    statement3
```

where `list` is a list of items:

- stated explicitly (e.g. `[1,2,3]`) or
- generated by a function, e.g. `range()`.

Today's Topics



- For-loops
- **range()**
- Variables
- Characters
- Strings

More on range():

```
1 #Predict what will be printed:
2
3 for num in [2,4,6,8,10]:
4     print(num)
5
6 sum = 0
7 for x in range(0,12,2):
8     print(x)
9     sum = sum + x
10
11 print(sum)
12
13 for c in "ABCD":
14     print(c)
```

Python Tutor

```
1 #Predict what will be printed:
2
3 for num in [2,4,6,8,10]:
4     print(num)
5
6 sum = 0
7 for x in range(0,12,2):
8     print(x)
9     sum = sum + x
10
11 print(sum)
12
13 for c in "ABCD":
14     print(c)
```

(Demo with pythonTutor)

range()

Simplest version:

- `range(stop)`



range()



Simplest version:

- `range(stop)`
- Produces a list: `[0,1,2,3,...,stop-1]`

range()



Simplest version:

- `range(stop)`
- Produces a list: `[0,1,2,3,...,stop-1]`
- For example, if you want the the list `[0,1,2,3,...,100]`, you would write:

range()



Simplest version:

- `range(stop)`
- Produces a list: `[0,1,2,3,...,stop-1]`
- For example, if you want the the list `[0,1,2,3,...,100]`, you would write:

```
range(101)
```

`range()`

What if you wanted to start somewhere else:



range()

What if you wanted to start somewhere else:

- `range(start, stop)`



range()



What if you wanted to start somewhere else:

- `range(start, stop)`
- Produces a list:
`[start, start+1, ..., stop-1]`

range()



What if you wanted to start somewhere else:

- `range(start, stop)`
- Produces a list:
`[start, start+1, ..., stop-1]`
- For example, if you want the the list
`[10, 11, ..., 20]`
you would write:

range()



What if you wanted to start somewhere else:

- `range(start, stop)`
- Produces a list:
`[start, start+1, ..., stop-1]`
- For example, if you want the the list
`[10, 11, ..., 20]`
you would write:

```
range(10, 21)
```

range()

What if you wanted to count by twos, or some other number:



range()

What if you wanted to count by twos, or some other number:

- `range(start, stop, step)`



range()

What if you wanted to count by twos, or some other number:

- `range(start, stop, step)`
- Produces a list:
`[start, start+step, start+2*step..., last]`
(where last is the largest $start+k*step$ less than stop)



range()



What if you wanted to count by twos, or some other number:

- `range(start, stop, step)`
- Produces a list:
`[start, start+step, start+2*step..., last]`
(where last is the largest $start+k*step$ less than stop)
- For example, if you want the the list `[5,10,...,50]` you would write:

range()

What if you wanted to count by twos, or some other number:

- `range(start, stop, step)`
- Produces a list:
`[start, start+step, start+2*step..., last]`
(where last is the largest $\text{start} + k * \text{step}$ less than stop)
- For example, if you want the the list `[5, 10, ..., 50]` you would write:

```
range(5, 51, 5)
```



In summary: `range()`



The three versions:

In summary: `range()`



The three versions:

- `range(stop)`

In summary: `range()`



The three versions:

- `range(stop)`
- `range(start, stop)`

In summary: `range()`



The three versions:

- `range(stop)`
- `range(start, stop)`
- `range(start, stop, step)`

Today's Topics



- For-loops
- `range()`
- **Variables**
- Characters
- Strings

Variables

- A **variable** is a reserved memory location for storing a value.



Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
 - ▶ **int**: integer or whole numbers



Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
 - ▶ **int**: integer or whole numbers
 - ▶ **float**: floating point or real numbers



Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
 - ▶ **int**: integer or whole numbers
 - ▶ **float**: floating point or real numbers
 - ▶ **string**: sequence of characters



Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
 - ▶ **int**: integer or whole numbers
 - ▶ **float**: floating point or real numbers
 - ▶ **string**: sequence of characters
 - ▶ **list**: a sequence of items



Variables

- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
 - ▶ **int**: integer or whole numbers
 - ▶ **float**: floating point or real numbers
 - ▶ **string**: sequence of characters
 - ▶ **list**: a sequence of items
e.g. [3, 1, 4, 5, 9] or
['violet', 'purple', 'indigo']



Variables



- A **variable** is a reserved memory location for storing a value.
- Different kinds, or **types**, of values need different amounts of space:
 - ▶ **int**: integer or whole numbers
 - ▶ **float**: floating point or real numbers
 - ▶ **string**: sequence of characters
 - ▶ **list**: a sequence of items
e.g. [3, 1, 4, 5, 9] or
['violet', 'purple', 'indigo']
 - ▶ **class variables**: for complex objects, like turtles.
- In Python (unlike other languages) you don't need to specify the type; it is deduced by its value.

Variable Names

- There's some rules about valid names for variables.



Variable Names



- There's some rules about valid names for variables.
- Can use the underscore ('_'), upper and lower case letters.

Variable Names



- There's some rules about valid names for variables.
- Can use the underscore ('_'), upper and lower case letters.
- Can also use numbers, just can't start a name with a number.

Variable Names



- There's some rules about valid names for variables.
- Can use the underscore ('_'), upper and lower case letters.
- Can also use numbers, just can't start a name with a number.
- Can't use symbols (like '+' or '*') since used for arithmetic.

Variable Names



- There's some rules about valid names for variables.
- Can use the underscore ('_'), upper and lower case letters.
- Can also use numbers, just can't start a name with a number.
- Can't use symbols (like '+' or '*') since used for arithmetic.
- Can't use some words that Python has reserved for itself (e.g. `for`).
(List of reserved words in *Think CS*, §2.5.)

Today's Topics



- For-loops
- `range()`
- Variables
- **Characters**
- Strings

Standardized Code for Characters

American Standard Code for Information Interchange (ASCII), 1960.

Standardized Code for Characters

American Standard Code for Information Interchange (ASCII), 1960.
(New version called: Unicode).

Standardized Code for Characters

American Standard Code for Information Interchange (ASCII), 1960.
(New version called: Unicode).

ASCII TABLE

| Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char |
|---------|-----|------------------------|---------|-----|---------|---------|-----|------|---------|-----|-------|
| 0 | 0 | [NULL] | 32 | 20 | [SPACE] | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 1 | [START OF HEADING] | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 2 | [START OF TEXT] | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 3 | [END OF TEXT] | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 4 | [END OF TRANSMISSION] | 36 | 24 | \$ | 68 | 44 | D | 100 | 64 | d |
| 5 | 5 | [ENQUIRY] | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 6 | [ACKNOWLEDGE] | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 7 | [BELL] | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 8 | [BACKSPACE] | 40 | 28 | (| 72 | 48 | H | 104 | 68 | h |
| 9 | 9 | [HORIZONTAL TAB] | 41 | 29 |) | 73 | 49 | I | 105 | 69 | i |
| 10 | A | [LINE FEED] | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | B | [VERTICAL TAB] | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | C | [FORM FEED] | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | D | [CARRIAGE RETURN] | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | E | [SHIFT OUT] | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | F | [SHIFT IN] | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | [DATA LINK ESCAPE] | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | [DEVICE CONTROL 1] | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | [DEVICE CONTROL 2] | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | [DEVICE CONTROL 3] | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | [DEVICE CONTROL 4] | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | [NEGATIVE ACKNOWLEDGE] | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | [SYNCHRONOUS IDLE] | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | [ENG OF TRANS. BLOCK] | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | [CANCEL] | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | [END OF MEDIUM] | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | [SUBSTITUTE] | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | [ESCAPE] | 59 | 3B | ; | 91 | 5B | [| 123 | 7B | { |
| 28 | 1C | [FILE SEPARATOR] | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | |
| 29 | 1D | [GROUP SEPARATOR] | 61 | 3D | = | 93 | 5D |] | 125 | 7D | } |
| 30 | 1E | [RECORD SEPARATOR] | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | [UNIT SEPARATOR] | 63 | 3F | ? | 95 | 5F | _ | 127 | 7F | [DEL] |

(wiki)

Converting from Character to Code:

(There is a link to the ASCII table on the course webpage, under 'Useful Links'.)

ASCII TABLE

| Decimal | Hex Char | Decimal | Hex Char | Decimal | Hex Char | Decimal | Hex Char |
|---------|----------|---------|----------|---------|----------|---------|----------|
| 0 | | 16 | 0x10 | 32 | @ | 48 | 0x30 |
| 1 | | 17 | 0x11 | 33 | A | 49 | 0x31 |
| 2 | | 18 | 0x12 | 34 | B | 50 | 0x32 |
| 3 | | 19 | 0x13 | 35 | C | 51 | 0x33 |
| 4 | | 20 | 0x14 | 36 | D | 52 | 0x34 |
| 5 | | 21 | 0x15 | 37 | E | 53 | 0x35 |
| 6 | | 22 | 0x16 | 38 | F | 54 | 0x36 |
| 7 | | 23 | 0x17 | 39 | | 55 | 0x37 |
| 8 | | 24 | 0x18 | 40 | | 56 | 0x38 |
| 9 | | 25 | 0x19 | 41 | a | 57 | 0x39 |
| 10 | | 26 | 0x1A | 42 | b | 58 | 0x3A |
| 11 | | 27 | 0x1B | 43 | c | 59 | 0x3B |
| 12 | | 28 | 0x1C | 44 | d | 60 | 0x3C |
| 13 | | 29 | 0x1D | 45 | e | 61 | 0x3D |
| 14 | | 30 | 0x1E | 46 | f | 62 | 0x3E |
| 15 | | 31 | 0x1F | 47 | g | 63 | 0x3F |
| 16 | 0x10 | 32 | @ | 64 | 0x40 | 80 | 0x50 |
| 17 | 0x11 | 33 | A | 65 | 0x41 | 81 | 0x51 |
| 18 | 0x12 | 34 | B | 66 | 0x42 | 82 | 0x52 |
| 19 | 0x13 | 35 | C | 67 | 0x43 | 83 | 0x53 |
| 20 | 0x14 | 36 | D | 68 | 0x44 | 84 | 0x54 |
| 21 | 0x15 | 37 | E | 69 | 0x45 | 85 | 0x55 |
| 22 | 0x16 | 38 | F | 70 | 0x46 | 86 | 0x56 |
| 23 | 0x17 | 39 | | 71 | 0x47 | 87 | 0x57 |
| 24 | 0x18 | 40 | | 72 | 0x48 | 88 | 0x58 |
| 25 | 0x19 | 41 | a | 73 | 0x49 | 89 | 0x59 |
| 26 | 0x1A | 42 | b | 74 | 0x4A | 90 | 0x5A |
| 27 | 0x1B | 43 | c | 75 | 0x4B | 91 | 0x5B |
| 28 | 0x1C | 44 | d | 76 | 0x4C | 92 | 0x5C |
| 29 | 0x1D | 45 | e | 77 | 0x4D | 93 | 0x5D |
| 30 | 0x1E | 46 | f | 78 | 0x4E | 94 | 0x5E |
| 31 | 0x1F | 47 | g | 79 | 0x4F | 95 | 0x5F |
| 32 | @ | 48 | 0x30 | 80 | 0x50 | 96 | 0x60 |
| 33 | A | 49 | 0x31 | 81 | 0x51 | 97 | 0x61 |
| 34 | B | 50 | 0x32 | 82 | 0x52 | 98 | 0x62 |
| 35 | C | 51 | 0x33 | 83 | 0x53 | 99 | 0x63 |
| 36 | D | 52 | 0x34 | 84 | 0x54 | 100 | 0x64 |
| 37 | E | 53 | 0x35 | 85 | 0x55 | 101 | 0x65 |
| 38 | F | 54 | 0x36 | 86 | 0x56 | 102 | 0x66 |
| 39 | | 55 | 0x37 | 87 | 0x57 | 103 | 0x67 |
| 40 | | 56 | 0x38 | 88 | 0x58 | 104 | 0x68 |
| 41 | a | 57 | 0x39 | 89 | 0x59 | 105 | 0x69 |
| 42 | b | 58 | 0x3A | 90 | 0x5A | 106 | 0x6A |
| 43 | c | 59 | 0x3B | 91 | 0x5B | 107 | 0x6B |
| 44 | d | 60 | 0x3C | 92 | 0x5C | 108 | 0x6C |
| 45 | e | 61 | 0x3D | 93 | 0x5D | 109 | 0x6D |
| 46 | f | 62 | 0x3E | 94 | 0x5E | 110 | 0x6E |
| 47 | g | 63 | 0x3F | 95 | 0x5F | 111 | 0x6F |
| 48 | 0x30 | 64 | 0x40 | 96 | 0x60 | 112 | 0x70 |
| 49 | 0x31 | 65 | 0x41 | 97 | 0x61 | 113 | 0x71 |
| 50 | 0x32 | 66 | 0x42 | 98 | 0x62 | 114 | 0x72 |
| 51 | 0x33 | 67 | 0x43 | 99 | 0x63 | 115 | 0x73 |
| 52 | 0x34 | 68 | 0x44 | 100 | 0x64 | 116 | 0x74 |
| 53 | 0x35 | 69 | 0x45 | 101 | 0x65 | 117 | 0x75 |
| 54 | 0x36 | 70 | 0x46 | 102 | 0x66 | 118 | 0x76 |
| 55 | 0x37 | 71 | 0x47 | 103 | 0x67 | 119 | 0x77 |
| 56 | 0x38 | 72 | 0x48 | 104 | 0x68 | 120 | 0x78 |
| 57 | 0x39 | 73 | 0x49 | 105 | 0x69 | 121 | 0x79 |
| 58 | 0x3A | 74 | 0x4A | 106 | 0x6A | 122 | 0x7A |
| 59 | 0x3B | 75 | 0x4B | 107 | 0x6B | 123 | 0x7B |
| 60 | 0x3C | 76 | 0x4C | 108 | 0x6C | 124 | 0x7C |
| 61 | 0x3D | 77 | 0x4D | 109 | 0x6D | 125 | 0x7D |
| 62 | 0x3E | 78 | 0x4E | 110 | 0x6E | 126 | 0x7E |
| 63 | 0x3F | 79 | 0x4F | 111 | 0x6F | 127 | 0x7F |
| 64 | 0x40 | 80 | 0x50 | 112 | 0x70 | 128 | 0x80 |
| 65 | 0x41 | 81 | 0x51 | 113 | 0x71 | 129 | 0x81 |
| 66 | 0x42 | 82 | 0x52 | 114 | 0x72 | 130 | 0x82 |
| 67 | 0x43 | 83 | 0x53 | 115 | 0x73 | 131 | 0x83 |
| 68 | 0x44 | 84 | 0x54 | 116 | 0x74 | 132 | 0x84 |
| 69 | 0x45 | 85 | 0x55 | 117 | 0x75 | 133 | 0x85 |
| 70 | 0x46 | 86 | 0x56 | 118 | 0x76 | 134 | 0x86 |
| 71 | 0x47 | 87 | 0x57 | 119 | 0x77 | 135 | 0x87 |
| 72 | 0x48 | 88 | 0x58 | 120 | 0x78 | 136 | 0x88 |
| 73 | 0x49 | 89 | 0x59 | 121 | 0x79 | 137 | 0x89 |
| 74 | 0x4A | 90 | 0x5A | 122 | 0x7A | 138 | 0x8A |
| 75 | 0x4B | 91 | 0x5B | 123 | 0x7B | 139 | 0x8B |
| 76 | 0x4C | 92 | 0x5C | 124 | 0x7C | 140 | 0x8C |
| 77 | 0x4D | 93 | 0x5D | 125 | 0x7D | 141 | 0x8D |
| 78 | 0x4E | 94 | 0x5E | 126 | 0x7E | 142 | 0x8E |
| 79 | 0x4F | 95 | 0x5F | 127 | 0x7F | 143 | 0x8F |
| 80 | 0x50 | 96 | 0x60 | 128 | 0x80 | 144 | 0x90 |
| 81 | 0x51 | 97 | 0x61 | 129 | 0x81 | 145 | 0x91 |
| 82 | 0x52 | 98 | 0x62 | 130 | 0x82 | 146 | 0x92 |
| 83 | 0x53 | 99 | 0x63 | 131 | 0x83 | 147 | 0x93 |
| 84 | 0x54 | 100 | 0x64 | 132 | 0x84 | 148 | 0x94 |
| 85 | 0x55 | 101 | 0x65 | 133 | 0x85 | 149 | 0x95 |
| 86 | 0x56 | 102 | 0x66 | 134 | 0x86 | 150 | 0x96 |
| 87 | 0x57 | 103 | 0x67 | 135 | 0x87 | 151 | 0x97 |
| 88 | 0x58 | 104 | 0x68 | 136 | 0x88 | 152 | 0x98 |
| 89 | 0x59 | 105 | 0x69 | 137 | 0x89 | 153 | 0x99 |
| 90 | 0x5A | 106 | 0x6A | 138 | 0x8A | 154 | 0x9A |
| 91 | 0x5B | 107 | 0x6B | 139 | 0x8B | 155 | 0x9B |
| 92 | 0x5C | 108 | 0x6C | 140 | 0x8C | 156 | 0x9C |
| 93 | 0x5D | 109 | 0x6D | 141 | 0x8D | 157 | 0x9D |
| 94 | 0x5E | 110 | 0x6E | 142 | 0x8E | 158 | 0x9E |
| 95 | 0x5F | 111 | 0x6F | 143 | 0x8F | 159 | 0x9F |
| 96 | 0x60 | 112 | 0x70 | 144 | 0x90 | 160 | 0xA0 |
| 97 | 0x61 | 113 | 0x71 | 145 | 0x91 | 161 | 0xA1 |
| 98 | 0x62 | 114 | 0x72 | 146 | 0x92 | 162 | 0xA2 |
| 99 | 0x63 | 115 | 0x73 | 147 | 0x93 | 163 | 0xA3 |
| 100 | 0x64 | 116 | 0x74 | 148 | 0x94 | 164 | 0xA4 |
| 101 | 0x65 | 117 | 0x75 | 149 | 0x95 | 165 | 0xA5 |
| 102 | 0x66 | 118 | 0x76 | 150 | 0x96 | 166 | 0xA6 |
| 103 | 0x67 | 119 | 0x77 | 151 | 0x97 | 167 | 0xA7 |
| 104 | 0x68 | 120 | 0x78 | 152 | 0x98 | 168 | 0xA8 |
| 105 | 0x69 | 121 | 0x79 | 153 | 0x99 | 169 | 0xA9 |
| 106 | 0x6A | 122 | 0x7A | 154 | 0x9A | 170 | 0xAA |
| 107 | 0x6B | 123 | 0x7B | 155 | 0x9B | 171 | 0xAB |
| 108 | 0x6C | 124 | 0x7C | 156 | 0x9C | 172 | 0xAC |
| 109 | 0x6D | 125 | 0x7D | 157 | 0x9D | 173 | 0xAD |
| 110 | 0x6E | 126 | 0x7E | 158 | 0x9E | 174 | 0xAE |
| 111 | 0x6F | 127 | 0x7F | 159 | 0x9F | 175 | 0xAF |
| 112 | 0x70 | 128 | 0x80 | 160 | 0xA0 | 176 | 0xB0 |
| 113 | 0x71 | 129 | 0x81 | 161 | 0xA1 | 177 | 0xB1 |
| 114 | 0x72 | 130 | 0x82 | 162 | 0xA2 | 178 | 0xB2 |
| 115 | 0x73 | 131 | 0x83 | 163 | 0xA3 | 179 | 0xB3 |
| 116 | 0x74 | 132 | 0x84 | 164 | 0xA4 | 180 | 0xB4 |
| 117 | 0x75 | 133 | 0x85 | 165 | 0xA5 | 181 | 0xB5 |
| 118 | 0x76 | 134 | 0x86 | 166 | 0xA6 | 182 | 0xB6 |
| 119 | 0x77 | 135 | 0x87 | 167 | 0xA7 | 183 | 0xB7 |
| 120 | 0x78 | 136 | 0x88 | 168 | 0xA8 | 184 | 0xB8 |
| 121 | 0x79 | 137 | 0x89 | 169 | 0xA9 | 185 | 0xB9 |
| 122 | 0x7A | 138 | 0x8A | 170 | 0xAA | 186 | 0xBA |
| 123 | 0x7B | 139 | 0x8B | 171 | 0xAB | 187 | 0xBB |
| 124 | 0x7C | 140 | 0x8C | 172 | 0xAC | 188 | 0xBC |
| 125 | 0x7D | 141 | 0x8D | 173 | 0xAD | 189 | 0xBD |
| 126 | 0x7E | 142 | 0x8E | 174 | 0xAE | 190 | 0xBE |
| 127 | 0x7F | 143 | 0x8F | 175 | 0xAF | 191 | 0xBF |
| 128 | 0x80 | 144 | 0x90 | 176 | 0xB0 | 192 | 0xC0 |
| 129 | 0x81 | 145 | 0x91 | 177 | 0xB1 | 193 | 0xC1 |
| 130 | 0x82 | 146 | 0x92 | 178 | 0xB2 | 194 | 0xC2 |
| 131 | 0x83 | 147 | 0x93 | 179 | 0xB3 | 195 | 0xC3 |
| 132 | 0x84 | 148 | 0x94 | 180 | 0xB4 | 196 | 0xC4 |
| 133 | 0x85 | 149 | 0x95 | 181 | 0xB5 | 197 | 0xC5 |
| 134 | 0x86 | 150 | 0x96 | 182 | 0xB6 | 198 | 0xC6 |
| 135 | 0x87 | 151 | 0x97 | 183 | 0xB7 | 199 | 0xC7 |
| 136 | 0x88 | 152 | 0x98 | 184 | 0xB8 | 200 | 0xC8 |
| 137 | 0x89 | 153 | 0x99 | 185 | 0xB9 | 201 | 0xC9 |
| 138 | 0x8A | 154 | 0x9A | 186 | 0xBA | 202 | 0xCA |
| 139 | 0x8B | 155 | 0x9B | 187 | 0xBB | 203 | 0xCB |
| 140 | 0x8C | 156 | 0x9C | 188 | 0xBC | 204 | 0xCC |
| 141 | 0x8D | 157 | 0x9D | 189 | 0xBD | 205 | 0xCD |
| 142 | 0x8E | 158 | 0x9E | 190 | 0xBE | 206 | 0xCE |
| 143 | 0x8F | 159 | 0x9F | 191 | 0xBF | 207 | 0xCF |
| 144 | 0x90 | 160 | 0xA0 | 192 | 0xC0 | 208 | 0xD0 |
| 145 | 0x91 | 161 | 0xA1 | 193 | 0xC1 | 209 | 0xD1 |
| 146 | 0x92 | 162 | 0xA2 | 194 | 0xC2 | 210 | 0xD2 |
| 147 | 0x93 | 163 | 0xA3 | 195 | 0xC3 | 211 | 0xD3 |
| 148 | 0x94 | 164 | 0xA4 | 196 | 0xC4 | 212 | 0xD4 |
| 149 | 0x95 | 165 | 0xA5 | 197 | 0xC5 | 213 | 0xD5 |
| 150 | 0x96 | 166 | 0xA6 | 198 | 0xC6 | 214 | 0xD6 |
| 151 | 0x97 | 167 | 0xA7 | 199 | 0xC7 | 215 | 0xD7 |
| 152 | 0x98 | 168 | 0xA8 | 200 | 0xC8 | 216 | 0xD8 |
| 153 | 0x99 | 169 | 0xA9 | 201 | 0xC9 | 217 | 0xD9 |
| 154 | 0x9A | 170 | 0xAA | 202 | 0xCA | 218 | 0xDA |
| 155 | 0x9B | 171 | 0xAB | 203 | 0xCB | 219 | 0xDB |
| 156 | 0x9C | 172 | 0xAC | 204 | 0xCC | 220 | 0xDC |
| 157 | 0x9D | 173 | 0xAD | 205 | 0xCD | 221 | 0xDD |
| 158 | 0x9E | 174 | 0xAE | 206 | 0xCE | 222 | 0xDE |
| 159 | 0x9F | 175 | 0xAF | 207 | 0xCF | 223 | 0xDF |
| 160 | 0xA0 | 176 | 0xB0 | 208 | 0xD0 | 224 | 0xE0 |
| 161 | 0xA1 | 177 | 0xB1 | 209 | 0xD1 | 225 | 0xE1 |
| 162 | 0xA2 | 178 | 0xB2 | 210 | 0xD2 | 226 | 0xE2 |
| 163 | 0xA3 | 179 | 0xB3 | 211 | 0xD3 | 227 | 0xE3 |
| 164 | 0xA4 | 180 | 0xB4 | 212 | 0xD4 | 228 | 0xE4 |
| 165 | 0xA5 | 181 | 0xB5 | 213 | 0xD5 | 229 | 0xE5 |
| 166 | 0xA6 | 182 | 0xB6 | 214 | 0xD6 | 230 | 0xE6 |
| 167 | 0xA7 | 183 | 0xB7 | 215 | 0xD7 | 231 | 0xE7 |
| 168 | 0xA8 | 184 | 0xB8 | 216 | 0xD8 | 232 | 0xE8 |
| 169 | 0xA9 | 185 | 0xB9 | 217 | 0xD9 | 233 | 0xE9 |
| 170 | 0xAA | 186 | 0xBA | 218 | 0xDA | 234 | 0xEA |
| 171 | 0xAB | 187 | 0xBB | 219 | 0xDB | 235 | 0xEB |
| 172 | 0xAC | 188 | 0xBC | 220 | 0xDC | 236 | 0xEC |
| 173 | 0xAD | 189 | 0xBD | 221 | 0xDD | 237 | 0xED |
| 174 | 0xAE | 190 | 0xBE | 222 | 0xDE | 238 | 0xEE |
| 175 | 0xAF | 191 | 0xBF | 223 | 0xDF | 239 | 0xEF |
| 176 | 0xB0 | 192 | 0xC0 | 224 | 0xE0 | 240 | 0xF0 |
| 177 | 0xB1 | 193 | 0xC1 | 225 | 0xE1 | 241 | 0xF1 |
| 178 | 0xB2 | 194 | 0xC2 | 226 | 0xE2 | 242 | 0xF2 |
| 179 | 0xB3 | 195 | 0xC3 | 227 | 0xE3 | 243 | 0xF3 |
| 180 | 0xB4 | 196 | 0xC4 | 228 | | | |

Converting from Character to Code:

(There is a link to the ASCII table on the course webpage, under 'Useful Links'.)

- `ord(c)`: returns Unicode (ASCII) of the character.

ASCII TABLE

| Decimal | Hex Char | Decimal | Hex Char | Decimal | Hex Char | Decimal | Hex Char |
|---------|----------|---------|----------|---------|----------|---------|----------|
| 0 | | 16 | | 32 | | 48 | |
| 1 | | 17 | | 33 | ! | 49 | 1 |
| 2 | | 18 | | 34 | " | 50 | 2 |
| 3 | | 19 | | 35 | # | 51 | 3 |
| 4 | | 20 | | 36 | \$ | 52 | 4 |
| 5 | | 21 | | 37 | % | 53 | 5 |
| 6 | | 22 | | 38 | & | 54 | 6 |
| 7 | | 23 | | 39 | ' | 55 | 7 |
| 8 | | 24 | | 40 | (| 56 | 8 |
| 9 | | 25 | | 41 |) | 57 | 9 |
| 10 | | 26 | | 42 | * | 58 | : |
| 11 | | 27 | | 43 | + | 59 | ; |
| 12 | | 28 | | 44 | , | 60 | < |
| 13 | | 29 | | 45 | - | 61 | = |
| 14 | | 30 | | 46 | . | 62 | > |
| 15 | | 31 | | 47 | / | 63 | ? |
| 16 | | 32 | ! | 64 | @ | 80 | P |
| 17 | | 33 | " | 65 | A | 81 | Q |
| 18 | | 34 | " | 66 | B | 82 | R |
| 19 | | 35 | # | 67 | C | 83 | S |
| 20 | | 36 | \$ | 68 | D | 84 | T |
| 21 | | 37 | % | 69 | E | 85 | U |
| 22 | | 38 | & | 70 | F | 86 | V |
| 23 | | 39 | ' | 71 | G | 87 | W |
| 24 | | 40 | (| 72 | H | 88 | X |
| 25 | | 41 |) | 73 | I | 89 | Y |
| 26 | | 42 | * | 74 | J | 90 | Z |
| 27 | | 43 | + | 75 | K | 91 | [|
| 28 | | 44 | , | 76 | L | 92 | \ |
| 29 | | 45 | - | 77 | M | 93 |] |
| 30 | | 46 | . | 78 | N | 94 | ^ |
| 31 | | 47 | / | 79 | O | 95 | _ |
| 32 | ! | 48 | 0 | 80 | P | 96 | ` |
| 33 | " | 49 | 1 | 81 | Q | 97 | a |
| 34 | " | 50 | 2 | 82 | R | 98 | b |
| 35 | # | 51 | 3 | 83 | S | 99 | c |
| 36 | \$ | 52 | 4 | 84 | T | 100 | d |
| 37 | % | 53 | 5 | 85 | U | 101 | e |
| 38 | & | 54 | 6 | 86 | V | 102 | f |
| 39 | ' | 55 | 7 | 87 | W | 103 | g |
| 40 | (| 56 | 8 | 88 | X | 104 | h |
| 41 |) | 57 | 9 | 89 | Y | 105 | i |
| 42 | * | 58 | : | 90 | Z | 106 | j |
| 43 | + | 59 | ; | 91 | [| 107 | k |
| 44 | , | 60 | < | 92 | \ | 108 | l |
| 45 | - | 61 | = | 93 |] | 109 | m |
| 46 | . | 62 | > | 94 | ^ | 110 | n |
| 47 | / | 63 | ? | 95 | _ | 111 | o |
| 48 | 0 | 64 | @ | 96 | ` | 112 | p |
| 49 | 1 | 65 | A | 97 | a | 113 | q |
| 50 | 2 | 66 | B | 98 | b | 114 | r |
| 51 | 3 | 67 | C | 99 | c | 115 | s |
| 52 | 4 | 70 | F | 100 | d | 116 | t |
| 53 | 5 | 71 | G | 101 | e | 117 | u |
| 54 | 6 | 72 | H | 102 | f | 118 | v |
| 55 | 7 | 73 | I | 103 | g | 119 | w |
| 56 | 8 | 74 | J | 104 | h | 120 | x |
| 57 | 9 | 75 | K | 105 | i | 121 | y |
| 58 | : | 76 | L | 106 | j | 122 | z |
| 59 | ; | 77 | M | 107 | k | 123 | { |
| 60 | < | 78 | N | 108 | l | 124 | |
| 61 | = | 79 | O | 109 | m | 125 | } |
| 62 | > | 80 | P | 110 | n | 126 | ~ |
| 63 | ? | 81 | Q | 111 | o | 127 | |
| 64 | @ | 82 | R | 112 | p | | |
| 65 | A | 83 | S | 113 | q | | |
| 66 | B | 84 | T | 114 | r | | |
| 67 | C | 85 | U | 115 | s | | |
| 68 | D | 86 | V | 116 | t | | |
| 69 | E | 87 | W | 117 | u | | |
| 70 | F | 88 | X | 118 | v | | |
| 71 | G | 89 | Y | 119 | w | | |
| 72 | H | 90 | Z | 120 | x | | |
| 73 | I | 91 | [| 121 | y | | |
| 74 | J | 92 | \ | 122 | z | | |
| 75 | K | 93 |] | | | | |
| 76 | L | 94 | ^ | | | | |
| 77 | M | 95 | _ | | | | |
| 78 | N | 96 | ` | | | | |
| 79 | O | 97 | a | | | | |
| 80 | P | 98 | b | | | | |
| 81 | Q | 99 | c | | | | |
| 82 | R | 100 | d | | | | |
| 83 | S | 101 | e | | | | |
| 84 | T | 102 | f | | | | |
| 85 | U | 103 | g | | | | |
| 86 | V | 104 | h | | | | |
| 87 | W | 105 | i | | | | |
| 88 | X | 106 | j | | | | |
| 89 | Y | 107 | k | | | | |
| 90 | Z | 108 | l | | | | |
| 91 | [| 109 | m | | | | |
| 92 | \ | 110 | n | | | | |
| 93 |] | 111 | o | | | | |
| 94 | ^ | 112 | p | | | | |
| 95 | _ | 113 | q | | | | |
| 96 | ` | 114 | r | | | | |
| 97 | a | 115 | s | | | | |
| 98 | b | 116 | t | | | | |
| 99 | c | 117 | u | | | | |
| 100 | d | 118 | v | | | | |
| 101 | e | 119 | w | | | | |
| 102 | f | 120 | x | | | | |
| 103 | g | 121 | y | | | | |
| 104 | h | 122 | z | | | | |
| 105 | i | | | | | | |
| 106 | j | | | | | | |
| 107 | k | | | | | | |
| 108 | l | | | | | | |
| 109 | m | | | | | | |
| 110 | n | | | | | | |
| 111 | o | | | | | | |
| 112 | p | | | | | | |
| 113 | q | | | | | | |
| 114 | r | | | | | | |
| 115 | s | | | | | | |
| 116 | t | | | | | | |
| 117 | u | | | | | | |
| 118 | v | | | | | | |
| 119 | w | | | | | | |
| 120 | x | | | | | | |
| 121 | y | | | | | | |
| 122 | z | | | | | | |
| 123 | { | | | | | | |
| 124 | | | | | | | |
| 125 | } | | | | | | |
| 126 | ~ | | | | | | |
| 127 | | | | | | | |

Converting from Character to Code:

(There is a link to the ASCII table on the course webpage, under 'Useful Links'.)

ASCII TABLE

| Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char |
|---------|-----|------|---------|-----|------|---------|-----|------|---------|-----|------|
| 0 | 00 | | 16 | 10 | P | 32 | 20 | R | 48 | 30 | T |
| 1 | 01 | SOH | 17 | 11 | Q | 33 | 21 | S | 49 | 31 | U |
| 2 | 02 | STX | 18 | 12 | R | 34 | 22 | T | 50 | 32 | V |
| 3 | 03 | ETX | 19 | 13 | S | 35 | 23 | U | 51 | 33 | W |
| 4 | 04 | END | 20 | 14 | T | 36 | 24 | V | 52 | 34 | X |
| 5 | 05 | SO | 21 | 15 | U | 37 | 25 | W | 53 | 35 | Y |
| 6 | 06 | SI | 22 | 16 | V | 38 | 26 | X | 54 | 36 | Z |
| 7 | 07 | BS | 23 | 17 | W | 39 | 27 | Y | 55 | 37 | [|
| 8 | 08 | HT | 24 | 18 | X | 40 | 28 | Z | 56 | 38 | \ |
| 9 | 09 | LF | 25 | 19 | Y | 41 | 29 | [| 57 | 39 |] |
| 10 | 0A | VT | 26 | 1A | Z | 42 | 2A | \ | 58 | 3A | ^ |
| 11 | 0B | FF | 27 | 1B | [| 43 | 2B |] | 59 | 3B | _ |
| 12 | 0C | DEL | 28 | 1C | \ | 44 | 2C | ^ | 60 | 3C | ` |
| 13 | 0D | | 29 | 1D |] | 45 | 2D | ^ | 61 | 3D | a |
| 14 | 0E | | 30 | 1E | ^ | 46 | 2E |] | 62 | 3E | b |
| 15 | 0F | | 31 | 1F | _ | 47 | 2F | ^ | 63 | 3F | c |
| 16 | 10 | SP | 32 | 20 | R | 48 | 30 | T | 64 | 40 | d |
| 17 | 11 | P | 33 | 21 | S | 49 | 31 | U | 65 | 41 | e |
| 18 | 12 | Q | 34 | 22 | T | 50 | 32 | V | 66 | 42 | f |
| 19 | 13 | R | 35 | 23 | U | 51 | 33 | W | 67 | 43 | g |
| 20 | 14 | S | 36 | 24 | V | 52 | 34 | X | 68 | 44 | h |
| 21 | 15 | T | 37 | 25 | W | 53 | 35 | Y | 69 | 45 | i |
| 22 | 16 | U | 38 | 26 | X | 54 | 36 | Z | 70 | 46 | j |
| 23 | 17 | V | 39 | 27 | Y | 55 | 37 | [| 71 | 47 | k |
| 24 | 18 | W | 40 | 28 | Z | 56 | 38 | \ | 72 | 48 | l |
| 25 | 19 | X | 41 | 29 | [| 57 | 39 |] | 73 | 49 | m |
| 26 | 1A | Z | 42 | 2A | \ | 58 | 3A | ^ | 74 | 4A | n |
| 27 | 1B | [| 43 | 2B |] | 59 | 3B | _ | 75 | 4B | o |
| 28 | 1C | \ | 44 | 2C | ^ | 60 | 3C | ` | 76 | 4C | p |
| 29 | 1D |] | 45 | 2D | ^ | 61 | 3D | a | 77 | 4D | q |
| 30 | 1E | ^ | 46 | 2E |] | 62 | 3E | b | 78 | 4E | r |
| 31 | 1F | _ | 47 | 2F | ^ | 63 | 3F | c | 79 | 4F | s |
| 32 | 20 | R | 48 | 30 | T | 64 | 40 | d | 80 | 50 | t |
| 33 | 21 | S | 49 | 31 | U | 65 | 41 | e | 81 | 51 | u |
| 34 | 22 | T | 50 | 32 | V | 66 | 42 | f | 82 | 52 | v |
| 35 | 23 | U | 51 | 33 | W | 67 | 43 | g | 83 | 53 | w |
| 36 | 24 | V | 52 | 34 | X | 68 | 44 | h | 84 | 54 | x |
| 37 | 25 | W | 53 | 35 | Y | 69 | 45 | i | 85 | 55 | y |
| 38 | 26 | X | 54 | 36 | Z | 70 | 46 | j | 86 | 56 | z |
| 39 | 27 | Y | 55 | 37 | [| 71 | 47 | k | 87 | 57 | { |
| 40 | 28 | Z | 56 | 38 | \ | 72 | 48 | l | 88 | 58 | } |
| 41 | 29 | [| 57 | 39 |] | 73 | 49 | m | 89 | 59 | ~ |
| 42 | 2A | \ | 58 | 3A | ^ | 90 | 5A | | 91 | 5B | |
| 43 | 2B |] | 59 | 3B | _ | 92 | 5C | | 92 | 5C | |
| 44 | 2C | ^ | 60 | 3C | ` | 93 | 5D | | 93 | 5D | |
| 45 | 2D | ^ | 61 | 3D | a | 94 | 5E | | 94 | 5E | |
| 46 | 2E |] | 62 | 3E | b | 95 | 5F | | 95 | 5F | |
| 47 | 2F | ^ | 63 | 3F | c | 96 | 60 | | 96 | 60 | |
| 48 | 30 | T | 64 | 40 | d | 97 | 61 | | 97 | 61 | |
| 49 | 31 | U | 65 | 41 | e | 98 | 62 | | 98 | 62 | |
| 50 | 32 | V | 66 | 42 | f | 99 | 63 | | 99 | 63 | |
| 51 | 33 | W | 67 | 43 | g | 100 | 64 | | 100 | 64 | |
| 52 | 34 | X | 68 | 44 | h | 101 | 65 | | 101 | 65 | |
| 53 | 35 | Y | 69 | 45 | i | 102 | 66 | | 102 | 66 | |
| 54 | 36 | Z | 70 | 46 | j | 103 | 67 | | 103 | 67 | |
| 55 | 37 | [| 71 | 47 | k | 104 | 68 | | 104 | 68 | |
| 56 | 38 | \ | 72 | 48 | l | 105 | 69 | | 105 | 69 | |
| 57 | 39 |] | 73 | 49 | m | 106 | 6A | | 106 | 6A | |
| 58 | 3A | ^ | 74 | 4A | n | 107 | 6B | | 107 | 6B | |
| 59 | 3B | _ | 75 | 4B | o | 108 | 6C | | 108 | 6C | |
| 60 | 3C | ` | 76 | 4C | p | 109 | 6D | | 109 | 6D | |
| 61 | 3D | a | 77 | 4D | q | 110 | 6E | | 110 | 6E | |
| 62 | 3E | b | 78 | 4E | r | 111 | 6F | | 111 | 6F | |
| 63 | 3F | c | 79 | 4F | s | 112 | 70 | | 112 | 70 | |
| 64 | 40 | d | 80 | 50 | t | 113 | 71 | | 113 | 71 | |
| 65 | 41 | e | 81 | 51 | u | 114 | 72 | | 114 | 72 | |
| 66 | 42 | f | 82 | 52 | v | 115 | 73 | | 115 | 73 | |
| 67 | 43 | g | 83 | 53 | w | 116 | 74 | | 116 | 74 | |
| 68 | 44 | h | 84 | 54 | x | 117 | 75 | | 117 | 75 | |
| 69 | 45 | i | 85 | 55 | y | 118 | 76 | | 118 | 76 | |
| 70 | 46 | j | 86 | 56 | z | 119 | 77 | | 119 | 77 | |
| 71 | 47 | k | 87 | 57 | { | 120 | 78 | | 120 | 78 | |
| 72 | 48 | l | 88 | 58 | } | 121 | 79 | | 121 | 79 | |
| 73 | 49 | m | 89 | 59 | ~ | 122 | 7A | | 122 | 7A | |
| 74 | 4A | n | 90 | 5A | | 123 | 7B | | 123 | 7B | |
| 75 | 4B | o | 91 | 5B | | 124 | 7C | | 124 | 7C | |
| 76 | 4C | p | 92 | 5C | | 125 | 7D | | 125 | 7D | |
| 77 | 4D | q | 93 | 5D | | 126 | 7E | | 126 | 7E | |
| 78 | 4E | r | 94 | 5E | | 127 | 7F | | 127 | 7F | |
| 79 | 4F | s | 95 | 5F | | 128 | 80 | | 128 | 80 | |
| 80 | 50 | t | 96 | 60 | | 129 | 81 | | 129 | 81 | |
| 81 | 51 | u | 97 | 61 | | 130 | 82 | | 130 | 82 | |
| 82 | 52 | v | 98 | 62 | | 131 | 83 | | 131 | 83 | |
| 83 | 53 | w | 99 | 63 | | 132 | 84 | | 132 | 84 | |
| 84 | 54 | x | 100 | 64 | | 133 | 85 | | 133 | 85 | |
| 85 | 55 | y | 101 | 65 | | 134 | 86 | | 134 | 86 | |
| 86 | 56 | z | 102 | 66 | | 135 | 87 | | 135 | 87 | |
| 87 | 57 | { | 103 | 67 | | 136 | 88 | | 136 | 88 | |
| 88 | 58 | } | 104 | 68 | | 137 | 89 | | 137 | 89 | |
| 89 | 59 | ~ | 105 | 69 | | 138 | 8A | | 138 | 8A | |
| 90 | 5A | | 106 | 6A | | 139 | 8B | | 139 | 8B | |
| 91 | 5B | | 107 | 6B | | 140 | 8C | | 140 | 8C | |
| 92 | 5C | | 108 | 6C | | 141 | 8D | | 141 | 8D | |
| 93 | 5D | | 109 | 6D | | 142 | 8E | | 142 | 8E | |
| 94 | 5E | | 110 | 6E | | 143 | 8F | | 143 | 8F | |
| 95 | 5F | | 111 | 6F | | 144 | 90 | | 144 | 90 | |
| 96 | 60 | | 112 | 70 | | 145 | 91 | | 145 | 91 | |
| 97 | 61 | | 113 | 71 | | 146 | 92 | | 146 | 92 | |
| 98 | 62 | | 114 | 72 | | 147 | 93 | | 147 | 93 | |
| 99 | 63 | | 115 | 73 | | 148 | 94 | | 148 | 94 | |
| 100 | 64 | | 116 | 74 | | 149 | 95 | | 149 | 95 | |
| 101 | 65 | | 117 | 75 | | 150 | 96 | | 150 | 96 | |
| 102 | 66 | | 118 | 76 | | 151 | 97 | | 151 | 97 | |
| 103 | 67 | | 119 | 77 | | 152 | 98 | | 152 | 98 | |
| 104 | 68 | | 120 | 78 | | 153 | 99 | | 153 | 99 | |
| 105 | 69 | | 121 | 79 | | 154 | 9A | | 154 | 9A | |
| 106 | 6A | | 122 | 7A | | 155 | 9B | | 155 | 9B | |
| 107 | 6B | | 123 | 7B | | 156 | 9C | | 156 | 9C | |
| 108 | 6C | | 124 | 7C | | 157 | 9D | | 157 | 9D | |
| 109 | 6D | | 125 | 7D | | 158 | 9E | | 158 | 9E | |
| 110 | 6E | | 126 | 7E | | 159 | 9F | | 159 | 9F | |
| 111 | 6F | | 127 | 7F | | 160 | A0 | | 160 | A0 | |
| 112 | 70 | | 128 | 80 | | 161 | A1 | | 161 | A1 | |
| 113 | 71 | | 129 | 81 | | 162 | A2 | | 162 | A2 | |
| 114 | 72 | | 130 | 82 | | 163 | A3 | | 163 | A3 | |
| 115 | 73 | | 131 | 83 | | 164 | A4 | | 164 | A4 | |
| 116 | 74 | | 132 | 84 | | 165 | A5 | | 165 | A5 | |
| 117 | 75 | | 133 | 85 | | 166 | A6 | | 166 | A6 | |
| 118 | 76 | | 134 | 86 | | 167 | A7 | | 167 | A7 | |
| 119 | 77 | | 135 | 87 | | 168 | A8 | | 168 | A8 | |
| 120 | 78 | | 136 | 88 | | 169 | A9 | | 169 | A9 | |
| 121 | 79 | | 137 | 89 | | 170 | AA | | 170 | AA | |
| 122 | 7A | | 138 | 8A | | 171 | AB | | 171 | AB | |
| 123 | 7B | | 139 | 8B | | 172 | AC | | 172 | AC | |
| 124 | 7C | | 140 | 8C | | 173 | AD | | 173 | AD | |
| 125 | 7D | | 141 | 8D | | 174 | AE | | 174 | AE | |
| 126 | 7E | | 142 | 8E | | 175 | AF | | 175 | AF | |
| 127 | 7F | | 143 | 8F | | 176 | B0 | | 176 | B0 | |
| 128 | 80 | | 144 | 90 | | 177 | B1 | | 177 | B1 | |
| 129 | 81 | | 145 | 91 | | 178 | B2 | | 178 | B2 | |
| 130 | 82 | | 146 | 92 | | 179 | B3 | | 179 | B3 | |
| 131 | 83 | | 147 | 93 | | 180 | B4 | | 180 | B4 | |
| 132 | 84 | | 148 | 94 | | 181 | B5 | | 181 | B5 | |
| 133 | 85 | | 149 | 95 | | 182 | B6 | | 182 | B6 | |
| 134 | 86 | | 150 | 96 | | 183 | B7 | | 183 | B7 | |
| 135 | 87 | | 151 | 97 | | 184 | B8 | | 184 | B8 | |
| 136 | 88 | | 152 | 98 | | 185 | B9 | | 185 | B9 | |
| 137 | 89 | | 153 | 99 | | 186 | BA | | 186 | BA | |
| 138 | 8A | | 154 | 9A | | 187 | BB | | 187 | BB | |
| 139 | 8B | | 155 | 9B | | 188 | BC | | 188 | BC | |
| 140 | 8C | | 156 | 9C | | 189 | BD | | 189 | BD | |
| 141 | 8D | | 157 | 9D | | 190 | BE | | 190 | BE | |
| 142 | 8E | | 158 | 9E | | 191 | BF | | 191 | BF | |
| 143 | 8F | | 159 | 9F | | 192 | C0 | | 192 | C0 | |
| 144 | 90 | | 160 | A0 | | 193 | C1 | | 193 | C1 | |
| 145 | 91 | | 161 | A1 | | 194 | C2 | | 194 | C2 | |
| 146 | 92 | | 162 | A2 | | 195 | C3 | | 195 | C3 | |
| 147 | 93 | | 163 | A3 | | 196 | C4 | | 196 | C4 | |
| 148 | 94 | | 164 | A4 | | 197 | C5 | | 197 | C5 | |
| 149 | 95 | | 165 | A5 | | 198 | C6 | | 198 | C6 | |
| 150 | 96 | | 166 | A6 | | 199 | C7 | | 199 | C7 | |
| 151 | 97 | | 167 | A7 | | 200 | C8 | | 200 | C8 | |
| 152 | 98 | | 168 | A8 | | 201 | | | | | |

Converting from Character to Code:

(There is a link to the ASCII table on the course webpage, under 'Useful Links'.)

ASCII TABLE

| Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char |
|---------|-----|------|---------|-----|------|---------|-----|------|---------|-----|------|
| 0 | 00 | NUL | 128 | 80 | DEL | 129 | 81 | | 130 | 82 | |
| 1 | 01 | | 129 | 81 | | 130 | 82 | | 131 | 83 | |
| 2 | 02 | | 130 | 82 | | 131 | 83 | | 132 | 84 | |
| 3 | 03 | | 131 | 83 | | 132 | 84 | | 133 | 85 | |
| 4 | 04 | | 132 | 84 | | 133 | 85 | | 134 | 86 | |
| 5 | 05 | | 133 | 85 | | 134 | 86 | | 135 | 87 | |
| 6 | 06 | | 134 | 86 | | 135 | 87 | | 136 | 88 | |
| 7 | 07 | | 135 | 87 | | 136 | 88 | | 137 | 89 | |
| 8 | 08 | | 136 | 88 | | 137 | 89 | | 138 | 8A | |
| 9 | 09 | | 137 | 89 | | 138 | 8A | | 139 | 8B | |
| 10 | 0A | | 138 | 8A | | 139 | 8B | | 140 | 8C | |
| 11 | 0B | | 139 | 8B | | 140 | 8C | | 141 | 8D | |
| 12 | 0C | | 140 | 8C | | 141 | 8D | | 142 | 8E | |
| 13 | 0D | | 141 | 8D | | 142 | 8E | | 143 | 8F | |
| 14 | 0E | | 142 | 8E | | 143 | 8F | | 144 | 90 | |
| 15 | 0F | | 143 | 8F | | 144 | 90 | | 145 | 91 | |
| 16 | 10 | | 144 | 90 | | 145 | 91 | | 146 | 92 | |
| 17 | 11 | | 145 | 91 | | 146 | 92 | | 147 | 93 | |
| 18 | 12 | | 146 | 92 | | 147 | 93 | | 148 | 94 | |
| 19 | 13 | | 147 | 93 | | 148 | 94 | | 149 | 95 | |
| 20 | 14 | | 148 | 94 | | 149 | 95 | | 150 | 96 | |
| 21 | 15 | | 149 | 95 | | 150 | 96 | | 151 | 97 | |
| 22 | 16 | | 150 | 96 | | 151 | 97 | | 152 | 98 | |
| 23 | 17 | | 151 | 97 | | 152 | 98 | | 153 | 99 | |
| 24 | 18 | | 152 | 98 | | 153 | 99 | | 154 | 9A | |
| 25 | 19 | | 153 | 99 | | 154 | 9A | | 155 | 9B | |
| 26 | 1A | | 154 | 9A | | 155 | 9B | | 156 | 9C | |
| 27 | 1B | | 155 | 9B | | 156 | 9C | | 157 | 9D | |
| 28 | 1C | | 156 | 9C | | 157 | 9D | | 158 | 9E | |
| 29 | 1D | | 157 | 9D | | 158 | 9E | | 159 | 9F | |
| 30 | 1E | | 158 | 9E | | 159 | 9F | | 160 | A0 | |
| 31 | 1F | | 159 | 9F | | 160 | A0 | | 161 | A1 | |
| 32 | 20 | | 160 | A0 | | 161 | A1 | | 162 | A2 | |
| 33 | 21 | | 161 | A1 | | 162 | A2 | | 163 | A3 | |
| 34 | 22 | | 162 | A2 | | 163 | A3 | | 164 | A4 | |
| 35 | 23 | | 163 | A3 | | 164 | A4 | | 165 | A5 | |
| 36 | 24 | | 164 | A4 | | 165 | A5 | | 166 | A6 | |
| 37 | 25 | | 165 | A5 | | 166 | A6 | | 167 | A7 | |
| 38 | 26 | | 166 | A6 | | 167 | A7 | | 168 | A8 | |
| 39 | 27 | | 167 | A7 | | 168 | A8 | | 169 | A9 | |
| 40 | 28 | | 168 | A8 | | 169 | A9 | | 170 | AA | |
| 41 | 29 | | 169 | A9 | | 170 | AA | | 171 | AB | |
| 42 | 2A | | 170 | AA | | 171 | AB | | 172 | AC | |
| 43 | 2B | | 171 | AB | | 172 | AC | | 173 | AD | |
| 44 | 2C | | 172 | AC | | 173 | AD | | 174 | AE | |
| 45 | 2D | | 173 | AD | | 174 | AE | | 175 | AF | |
| 46 | 2E | | 174 | AE | | 175 | AF | | 176 | B0 | |
| 47 | 2F | | 175 | AF | | 176 | B0 | | 177 | B1 | |
| 48 | 30 | | 176 | B0 | | 177 | B1 | | 178 | B2 | |
| 49 | 31 | | 177 | B1 | | 178 | B2 | | 179 | B3 | |
| 50 | 32 | | 178 | B2 | | 179 | B3 | | 180 | B4 | |
| 51 | 33 | | 179 | B3 | | 180 | B4 | | 181 | B5 | |
| 52 | 34 | | 180 | B4 | | 181 | B5 | | 182 | B6 | |
| 53 | 35 | | 181 | B5 | | 182 | B6 | | 183 | B7 | |
| 54 | 36 | | 182 | B6 | | 183 | B7 | | 184 | B8 | |
| 55 | 37 | | 183 | B7 | | 184 | B8 | | 185 | B9 | |
| 56 | 38 | | 184 | B8 | | 185 | B9 | | 186 | BA | |
| 57 | 39 | | 185 | B9 | | 186 | BA | | 187 | BB | |
| 58 | 3A | | 186 | BA | | 187 | BB | | 188 | BC | |
| 59 | 3B | | 187 | BB | | 188 | BC | | 189 | BD | |
| 60 | 3C | | 188 | BC | | 189 | BD | | 190 | BE | |
| 61 | 3D | | 189 | BD | | 190 | BE | | 191 | BF | |
| 62 | 3E | | 190 | BE | | 191 | BF | | 192 | C0 | |
| 63 | 3F | | 191 | BF | | 192 | C0 | | 193 | C1 | |
| 64 | 40 | | 192 | C0 | | 193 | C1 | | 194 | C2 | |
| 65 | 41 | | 193 | C1 | | 194 | C2 | | 195 | C3 | |
| 66 | 42 | | 194 | C2 | | 195 | C3 | | 196 | C4 | |
| 67 | 43 | | 195 | C3 | | 196 | C4 | | 197 | C5 | |
| 68 | 44 | | 196 | C4 | | 197 | C5 | | 198 | C6 | |
| 69 | 45 | | 197 | C5 | | 198 | C6 | | 199 | C7 | |
| 70 | 46 | | 198 | C6 | | 199 | C7 | | 200 | C8 | |
| 71 | 47 | | 199 | C7 | | 200 | C8 | | 201 | C9 | |
| 72 | 48 | | 200 | C8 | | 201 | C9 | | 202 | CA | |
| 73 | 49 | | 201 | C9 | | 202 | CA | | 203 | CB | |
| 74 | 4A | | 202 | CA | | 203 | CB | | 204 | CC | |
| 75 | 4B | | 203 | CB | | 204 | CC | | 205 | CD | |
| 76 | 4C | | 204 | CC | | 205 | CD | | 206 | CE | |
| 77 | 4D | | 205 | CD | | 206 | CE | | 207 | CF | |
| 78 | 4E | | 206 | CE | | 207 | CF | | 208 | D0 | |
| 79 | 4F | | 207 | CF | | 208 | D0 | | 209 | D1 | |
| 80 | 50 | | 208 | D0 | | 209 | D1 | | 210 | D2 | |
| 81 | 51 | | 209 | D1 | | 210 | D2 | | 211 | D3 | |
| 82 | 52 | | 210 | D2 | | 211 | D3 | | 212 | D4 | |
| 83 | 53 | | 211 | D3 | | 212 | D4 | | 213 | D5 | |
| 84 | 54 | | 212 | D4 | | 213 | D5 | | 214 | D6 | |
| 85 | 55 | | 213 | D5 | | 214 | D6 | | 215 | D7 | |
| 86 | 56 | | 214 | D6 | | 215 | D7 | | 216 | D8 | |
| 87 | 57 | | 215 | D7 | | 216 | D8 | | 217 | D9 | |
| 88 | 58 | | 216 | D8 | | 217 | D9 | | 218 | DA | |
| 89 | 59 | | 217 | D9 | | 218 | DA | | 219 | DB | |
| 90 | 5A | | 218 | DA | | 219 | DB | | 220 | DC | |
| 91 | 5B | | 219 | DB | | 220 | DC | | 221 | DD | |
| 92 | 5C | | 220 | DC | | 221 | DD | | 222 | DE | |
| 93 | 5D | | 221 | DD | | 222 | DE | | 223 | DF | |
| 94 | 5E | | 222 | DE | | 223 | DF | | 224 | E0 | |
| 95 | 5F | | 223 | DF | | 224 | E0 | | 225 | E1 | |
| 96 | 60 | | 224 | E0 | | 225 | E1 | | 226 | E2 | |
| 97 | 61 | | 225 | E1 | | 226 | E2 | | 227 | E3 | |
| 98 | 62 | | 226 | E2 | | 227 | E3 | | 228 | E4 | |
| 99 | 63 | | 227 | E3 | | 228 | E4 | | 229 | E5 | |
| 100 | 64 | | 228 | E4 | | 229 | E5 | | 230 | E6 | |
| 101 | 65 | | 229 | E5 | | 230 | E6 | | 231 | E7 | |
| 102 | 66 | | 230 | E6 | | 231 | E7 | | 232 | E8 | |
| 103 | 67 | | 231 | E7 | | 232 | E8 | | 233 | E9 | |
| 104 | 68 | | 232 | E8 | | 233 | E9 | | 234 | EA | |
| 105 | 69 | | 233 | E9 | | 234 | EA | | 235 | EB | |
| 106 | 6A | | 234 | EA | | 235 | EB | | 236 | EC | |
| 107 | 6B | | 235 | EB | | 236 | EC | | 237 | ED | |
| 108 | 6C | | 236 | EC | | 237 | ED | | 238 | EE | |
| 109 | 6D | | 237 | ED | | 238 | EE | | 239 | EF | |
| 110 | 6E | | 238 | EE | | 239 | EF | | 240 | F0 | |
| 111 | 6F | | 239 | EF | | 240 | F0 | | 241 | F1 | |
| 112 | 70 | | 240 | F0 | | 241 | F1 | | 242 | F2 | |
| 113 | 71 | | 241 | F1 | | 242 | F2 | | 243 | F3 | |
| 114 | 72 | | 242 | F2 | | 243 | F3 | | 244 | F4 | |
| 115 | 73 | | 243 | F3 | | 244 | F4 | | 245 | F5 | |
| 116 | 74 | | 244 | F4 | | 245 | F5 | | 246 | F6 | |
| 117 | 75 | | 245 | F5 | | 246 | F6 | | 247 | F7 | |
| 118 | 76 | | 246 | F6 | | 247 | F7 | | 248 | F8 | |
| 119 | 77 | | 247 | F7 | | 248 | F8 | | 249 | F9 | |
| 120 | 78 | | 248 | F8 | | 249 | F9 | | 250 | FA | |
| 121 | 79 | | 249 | F9 | | 250 | FA | | 251 | FB | |
| 122 | 7A | | 250 | FA | | 251 | FB | | 252 | FC | |
| 123 | 7B | | 251 | FB | | 252 | FC | | 253 | FD | |
| 124 | 7C | | 252 | FC | | 253 | FD | | 254 | FE | |
| 125 | 7D | | 253 | FD | | 254 | FE | | 255 | FF | |
| 126 | 7E | | 254 | FE | | 255 | FF | | | | |
| 127 | 7F | | 255 | FF | | | | | | | |

- `ord(c)`: returns Unicode (ASCII) of the character.
- Example: `ord('a')` returns 97.
- `chr(x)`: returns the character whose Unicode is x.

Converting from Character to Code:

(There is a link to the ASCII table on the course webpage, under 'Useful Links'.)

ASCII TABLE

| Decimal | Hex Char | Decimal | Hex Char | Decimal | Hex Char | Decimal | Hex Char |
|---------|----------|---------|----------|---------|----------|---------|----------|
| 0 | | 16 | P | 32 | @ | 48 | 0 |
| 1 | SOH | 17 | Q | 33 | A | 49 | 1 |
| 2 | STX | 18 | R | 34 | B | 50 | 2 |
| 3 | ETX | 19 | S | 35 | C | 51 | 3 |
| 4 | END | 20 | T | 36 | D | 52 | 4 |
| 5 | SO | 21 | U | 37 | E | 53 | 5 |
| 6 | ST | 22 | V | 38 | F | 54 | 6 |
| 7 | HT | 23 | W | 39 | G | 55 | 7 |
| 8 | LF | 24 | X | 40 | H | 56 | 8 |
| 9 | VT | 25 | Y | 41 | I | 57 | 9 |
| 10 | FF | 26 | Z | 42 | J | 58 | : |
| 11 | | 27 | [| 43 | K | 59 | ; |
| 12 | | 28 | \ | 44 | L | 60 | < |
| 13 | CR | 29 |] | 45 | M | 61 | = |
| 14 | | 30 | ^ | 46 | N | 62 | > |
| 15 | | 31 | _ | 47 | O | 63 | ? |

- `ord(c)`: returns Unicode (ASCII) of the character.
- Example: `ord('a')` returns 97.
- `chr(x)`: returns the character whose Unicode is x.
- Example: `chr(97)` returns 'a'.

Converting from Character to Code:

(There is a link to the ASCII table on the course webpage, under 'Useful Links'.)

ASCII TABLE

| Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char |
|---------|-----|------|---------|-----|------|---------|-----|------|---------|-----|------|
| 0 | 00 | | 16 | 10 | P | 32 | 20 | R | 48 | 30 | T |
| 1 | 01 | SOH | 17 | 11 | Q | 33 | 21 | S | 49 | 31 | U |
| 2 | 02 | STX | 18 | 12 | R | 34 | 22 | T | 50 | 32 | V |
| 3 | 03 | ETX | 19 | 13 | S | 35 | 23 | U | 51 | 33 | W |
| 4 | 04 | END | 20 | 14 | T | 36 | 24 | V | 52 | 34 | X |
| 5 | 05 | SO | 21 | 15 | U | 37 | 25 | W | 53 | 35 | Y |
| 6 | 06 | ST | 22 | 16 | V | 38 | 26 | X | 54 | 36 | Z |
| 7 | 07 | HT | 23 | 17 | W | 39 | 27 | Y | 55 | 37 | [|
| 8 | 08 | LF | 24 | 18 | X | 40 | 28 | Z | 56 | 38 | \ |
| 9 | 09 | VT | 25 | 19 | Y | 41 | 29 | [| 57 | 39 |] |
| 10 | 0A | FF | 26 | 1A | Z | 42 | 2A | \ | 58 | 3A | ^ |
| 11 | 0B | SO | 27 | 1B | [| 43 | 2B |] | 59 | 3B | _ |
| 12 | 0C | ST | 28 | 1C | \ | 44 | 2C | ^ | 60 | 3C | ` |
| 13 | 0D | HT | 29 | 1D |] | 45 | 2D | _ | 61 | 3D | a |
| 14 | 0E | LF | 30 | 1E | ^ | 46 | 2E | ` | 62 | 3E | b |
| 15 | 0F | VT | 31 | 1F | _ | 47 | 2F | a | 63 | 3F | c |
| 16 | 10 | SO | 32 | 20 | P | 48 | 30 | t | 64 | 40 | d |
| 17 | 11 | ST | 33 | 21 | S | 49 | 31 | u | 65 | 41 | e |
| 18 | 12 | HT | 34 | 22 | T | 50 | 32 | v | 66 | 42 | f |
| 19 | 13 | LF | 35 | 23 | U | 51 | 33 | w | 67 | 43 | g |
| 20 | 14 | VT | 36 | 24 | V | 52 | 34 | x | 68 | 44 | h |
| 21 | 15 | SO | 37 | 25 | W | 53 | 35 | y | 69 | 45 | i |
| 22 | 16 | ST | 38 | 26 | X | 54 | 36 | z | 70 | 46 | j |
| 23 | 17 | HT | 39 | 27 | Y | 55 | 37 | [| 71 | 47 | k |
| 24 | 18 | LF | 40 | 28 | Z | 56 | 38 | \ | 72 | 48 | l |
| 25 | 19 | VT | 41 | 29 | [| 57 | 39 |] | 73 | 49 | m |
| 26 | 1A | SO | 42 | 2A | \ | 58 | 3A | ^ | 74 | 4A | n |
| 27 | 1B | ST | 43 | 2B |] | 59 | 3B | _ | 75 | 4B | o |
| 28 | 1C | HT | 44 | 2C | ^ | 60 | 3C | ` | 76 | 4C | p |
| 29 | 1D | LF | 45 | 2D | _ | 61 | 3D | a | 77 | 4D | q |
| 30 | 1E | VT | 46 | 2E | ` | 62 | 3E | b | 78 | 4E | r |
| 31 | 1F | SO | 47 | 2F | a | 63 | 3F | c | 79 | 4F | s |
| 32 | 20 | P | 48 | 30 | t | 64 | 40 | d | 80 | 50 | t |
| 33 | 21 | S | 49 | 31 | u | 65 | 41 | e | 81 | 51 | u |
| 34 | 22 | T | 50 | 32 | v | 66 | 42 | f | 82 | 52 | v |
| 35 | 23 | U | 51 | 33 | w | 67 | 43 | g | 83 | 53 | w |
| 36 | 24 | V | 52 | 34 | x | 68 | 44 | h | 84 | 54 | x |
| 37 | 25 | W | 53 | 35 | y | 69 | 45 | i | 85 | 55 | y |
| 38 | 26 | X | 54 | 36 | z | 70 | 46 | j | 86 | 56 | z |
| 39 | 27 | Y | 55 | 37 | [| 71 | 47 | k | 87 | 57 | [|
| 40 | 28 | Z | 56 | 38 | \ | 72 | 48 | l | 88 | 58 | \ |
| 41 | 29 | [| 57 | 39 |] | 73 | 49 | m | 89 | 59 |] |
| 42 | 2A | \ | 58 | 3A | ^ | 74 | 4A | n | 90 | 5A | ^ |
| 43 | 2B |] | 59 | 3B | _ | 75 | 4B | o | 91 | 5B | _ |
| 44 | 2C | ^ | 60 | 3C | ` | 76 | 4C | p | 92 | 5C | ` |
| 45 | 2D | _ | 61 | 3D | a | 77 | 4D | q | 93 | 5D | a |
| 46 | 2E | ` | 62 | 3E | b | 78 | 4E | r | 94 | 5E | b |
| 47 | 2F | a | 63 | 3F | c | 79 | 4F | s | 95 | 5F | c |
| 48 | 30 | t | 64 | 40 | d | 80 | 50 | t | 96 | 60 | d |
| 49 | 31 | u | 65 | 41 | e | 81 | 51 | u | 97 | 61 | e |
| 50 | 32 | v | 66 | 42 | f | 82 | 52 | v | 98 | 62 | f |
| 51 | 33 | w | 67 | 43 | g | 83 | 53 | w | 99 | 63 | g |
| 52 | 34 | x | 68 | 44 | h | 84 | 54 | x | 100 | 64 | h |
| 53 | 35 | y | 69 | 45 | i | 85 | 55 | y | 101 | 65 | i |
| 54 | 36 | z | 70 | 46 | j | 86 | 56 | z | 102 | 66 | j |
| 55 | 37 | [| 71 | 47 | k | 87 | 57 | [| 103 | 67 | k |
| 56 | 38 | \ | 72 | 48 | l | 88 | 58 | \ | 104 | 68 | l |
| 57 | 39 |] | 73 | 49 | m | 89 | 59 |] | 105 | 69 | m |
| 58 | 3A | ^ | 74 | 4A | n | 90 | 5A | ^ | 106 | 6A | n |
| 59 | 3B | _ | 75 | 4B | o | 91 | 5B | _ | 107 | 6B | o |
| 60 | 3C | ` | 76 | 4C | p | 92 | 5C | ` | 108 | 6C | p |
| 61 | 3D | a | 77 | 4D | q | 93 | 5D | a | 109 | 6D | q |
| 62 | 3E | b | 78 | 4E | r | 94 | 5E | b | 110 | 6E | r |
| 63 | 3F | c | 79 | 4F | s | 95 | 5F | c | 111 | 6F | s |
| 64 | 40 | d | 80 | 50 | t | 96 | 60 | d | 112 | 70 | t |
| 65 | 41 | e | 81 | 51 | u | 97 | 61 | e | 113 | 71 | u |
| 66 | 42 | f | 82 | 52 | v | 98 | 62 | f | 114 | 72 | v |
| 67 | 43 | g | 83 | 53 | w | 99 | 63 | g | 115 | 73 | w |
| 68 | 44 | h | 84 | 54 | x | 100 | 64 | h | 116 | 74 | x |
| 69 | 45 | i | 85 | 55 | y | 101 | 65 | i | 117 | 75 | y |
| 70 | 46 | j | 86 | 56 | z | 102 | 66 | j | 118 | 76 | z |
| 71 | 47 | k | 87 | 57 | [| 103 | 67 | k | 119 | 77 | [|
| 72 | 48 | l | 88 | 58 | \ | 104 | 68 | l | 120 | 78 | \ |
| 73 | 49 | m | 89 | 59 |] | 105 | 69 | m | 121 | 79 |] |
| 74 | 4A | n | 90 | 5A | ^ | 106 | 6A | n | 122 | 7A | ^ |
| 75 | 4B | o | 91 | 5B | _ | 107 | 6B | o | 123 | 7B | _ |
| 76 | 4C | p | 92 | 5C | ` | 108 | 6C | p | 124 | 7C | ` |
| 77 | 4D | q | 93 | 5D | a | 109 | 6D | q | 125 | 7D | a |
| 78 | 4E | r | 94 | 5E | b | 110 | 6E | r | 126 | 7E | b |
| 79 | 4F | s | 95 | 5F | c | 111 | 6F | s | 127 | 7F | c |
| 80 | 50 | t | 96 | 60 | d | 112 | 70 | t | | | |
| 81 | 51 | u | 97 | 61 | e | 113 | 71 | u | | | |
| 82 | 52 | v | 98 | 62 | f | 114 | 72 | v | | | |
| 83 | 53 | w | 99 | 63 | g | 115 | 73 | w | | | |
| 84 | 54 | x | 100 | 64 | h | 116 | 74 | x | | | |
| 85 | 55 | y | 101 | 65 | i | 117 | 75 | y | | | |
| 86 | 56 | z | 102 | 66 | j | 118 | 76 | z | | | |
| 87 | 57 | [| 103 | 67 | k | 119 | 77 | [| | | |
| 88 | 58 | \ | 104 | 68 | l | 120 | 78 | \ | | | |
| 89 | 59 |] | 105 | 69 | m | 121 | 79 |] | | | |
| 90 | 5A | ^ | 106 | 6A | n | 122 | 7A | ^ | | | |
| 91 | 5B | _ | 107 | 6B | o | | | | | | |
| 92 | 5C | ` | 108 | 6C | p | | | | | | |
| 93 | 5D | a | 109 | 6D | q | | | | | | |
| 94 | 5E | b | 110 | 6E | r | | | | | | |
| 95 | 5F | c | 111 | 6F | s | | | | | | |
| 96 | 60 | d | 112 | 70 | t | | | | | | |
| 97 | 61 | e | 113 | 71 | u | | | | | | |
| 98 | 62 | f | 114 | 72 | v | | | | | | |
| 99 | 63 | g | 115 | 73 | w | | | | | | |
| 100 | 64 | h | 116 | 74 | x | | | | | | |
| 101 | 65 | i | 117 | 75 | y | | | | | | |
| 102 | 66 | j | 118 | 76 | z | | | | | | |
| 103 | 67 | k | 119 | 77 | [| | | | | | |
| 104 | 68 | l | 120 | 78 | \ | | | | | | |
| 105 | 69 | m | 121 | 79 |] | | | | | | |
| 106 | 6A | n | 122 | 7A | ^ | | | | | | |
| 107 | 6B | o | | | | | | | | | |
| 108 | 6C | p | | | | | | | | | |
| 109 | 6D | q | | | | | | | | | |
| 110 | 6E | r | | | | | | | | | |
| 111 | 6F | s | | | | | | | | | |
| 112 | 70 | t | | | | | | | | | |
| 113 | 71 | u | | | | | | | | | |
| 114 | 72 | v | | | | | | | | | |
| 115 | 73 | w | | | | | | | | | |
| 116 | 74 | x | | | | | | | | | |
| 117 | 75 | y | | | | | | | | | |
| 118 | 76 | z | | | | | | | | | |
| 119 | 77 | [| | | | | | | | | |
| 120 | 78 | \ | | | | | | | | | |
| 121 | 79 |] | | | | | | | | | |
| 122 | 7A | ^ | | | | | | | | | |
| 123 | 7B | _ | | | | | | | | | |
| 124 | 7C | ` | | | | | | | | | |
| 125 | 7D | a | | | | | | | | | |
| 126 | 7E | b | | | | | | | | | |
| 127 | 7F | c | | | | | | | | | |

- `ord(c)`: returns Unicode (ASCII) of the character.
- Example: `ord('a')` returns 97.
- `chr(x)`: returns the character whose Unicode is x.
- Example: `chr(97)` returns 'a'.
- What is `chr(33)`?

In Pairs or Triples...

Some review and some novel challenges:

../images/csci127/caesarCipher.png

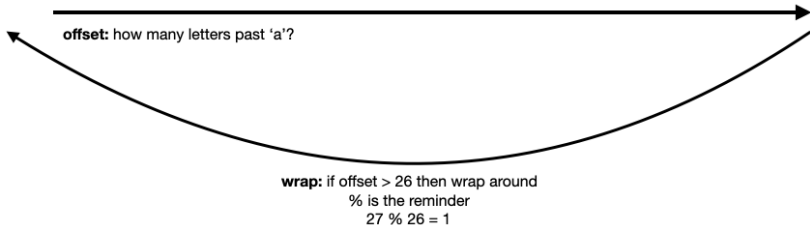
Python Tutor

```
1 #Predict what will be printed:
2
3 for c in range(65,90):
4     print(chr(c))
5
6 message = "I love Python"
7 newMessage = ""
8 for c in message:
9     print(ord(c))    #Print the Unicode of each number
10    print(chr(ord(c)+1))    #Print the next character
11    newMessage = newMessage + chr(ord(c)+1) #Add to the new message
12 print("The coded message is", newMessage)
13
14 word = "zebra"
15 codedWord = ""
16 for ch in word:
17     offset = ord(ch) - ord('a') + 1 #how many letters past 'a'
18     wrap = offset % 26 #if larger than 26, wrap back to 0
19     newChar = chr(ord('a') + wrap) #compute the new letter
20     print(wrap, chr(ord('a') + wrap))    #Print the wrap & new lett
21     codedWord = codedWord + newChar #add the newChar to the coded w
22
23 print("The coded word (with wrap) is", codedWord)
```

(Demo with pythonTutor)

Wrap

| | | | | | | | | | | | | | |
|-------|----|----|----|--|--|--|-----|--|--|--|-----|-----|-----|
| chr() | a | b | c | | | | ... | | | | x | y | z |
| ord() | 97 | 98 | 99 | | | | ... | | | | 120 | 121 | 122 |



User Input

Covered in detail in Lab 2:

../images/csci127/inputMessage.png

Side Note: '+' for numbers and strings



- `x = 3 + 5` stores the number 8 in memory location `x`.

Side Note: '+' for numbers and strings



- $x = 3 + 5$ stores the number 8 in memory location x .
- $x = x + 1$ increases x by 1.

Side Note: '+' for numbers and strings



- `x = 3 + 5` stores the number 8 in memory location `x`.
- `x = x + 1` increases `x` by 1.
- `s = "hi" + "Mom"` stores "hiMom" in memory locations `s`.

Side Note: '+' for numbers and strings



- `x = 3 + 5` stores the number 8 in memory location `x`.
- `x = x + 1` increases `x` by 1.
- `s = "hi" + "Mom"` stores "hiMom" in memory locations `s`.
- `s = s + "A"` adds the letter "A" to the end of the strings `s`.

Today's Topics



- For-loops
- `range()`
- Variables
- Characters
- **Strings**

More on Strings: String Methods

```
s = "FridaysSaturdaysSundays"  
num = s.count("s")
```

- The first line creates a variable, called `s`, that stores the string: "FridaysSaturdaysSundays"

More on Strings: String Methods

```
s = "FridaysSaturdaysSundays"  
num = s.count("s")
```

- The first line creates a variable, called `s`, that stores the string: "FridaysSaturdaysSundays"
- There are many useful functions for strings (more in Lab 2).

More on Strings: String Methods

```
s = "FridaysSaturdaysSundays"  
num = s.count("s")
```

- The first line creates a variable, called `s`, that stores the string: `"FridaysSaturdaysSundays"`
- There are many useful functions for strings (more in Lab 2).
- `s.count(x)` will count the number of times the pattern, `x`, appears in `s`.

More on Strings: String Methods

```
s = "FridaysSaturdaysSundays"  
num = s.count("s")
```

- The first line creates a variable, called `s`, that stores the string: "FridaysSaturdaysSundays"
- There are many useful functions for strings (more in Lab 2).
- `s.count(x)` will count the number of times the pattern, `x`, appears in `s`.
 - ▶ `s.count("s")` counts the number of lower case `s` that occurs.

More on Strings: String Methods

```
s = "FridaysSaturdaysSundays"  
num = s.count("s")
```

- The first line creates a variable, called `s`, that stores the string: "FridaysSaturdaysSundays"
- There are many useful functions for strings (more in Lab 2).
- `s.count(x)` will count the number of times the pattern, `x`, appears in `s`.
 - ▶ `s.count("s")` counts the number of lower case `s` that occurs.
 - ▶ `num = s.count("s")` stores the result in the variable `num`, for later.

More on Strings: String Methods

```
s = "FridaysSaturdaysSundays"  
num = s.count("s")
```

- The first line creates a variable, called `s`, that stores the string: `"FridaysSaturdaysSundays"`
- There are many useful functions for strings (more in Lab 2).
- `s.count(x)` will count the number of times the pattern, `x`, appears in `s`.
 - ▶ `s.count("s")` counts the number of lower case `s` that occurs.
 - ▶ `num = s.count("s")` stores the result in the variable `num`, for later.
 - ▶ What would `print(s.count("sS"))` output?

More on Strings: String Methods

```
s = "FridaysSaturdaysSundays"  
num = s.count("s")
```

- The first line creates a variable, called `s`, that stores the string: "FridaysSaturdaysSundays"
- There are many useful functions for strings (more in Lab 2).
- `s.count(x)` will count the number of times the pattern, `x`, appears in `s`.
 - ▶ `s.count("s")` counts the number of lower case `s` that occurs.
 - ▶ `num = s.count("s")` stores the result in the variable `num`, for later.
 - ▶ What would `print(s.count("sS"))` output?
 - ▶ What about:

```
mess = "10 20 21 9 101 35"  
mults = mess.count("0 ")  
print(mults)
```

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[0]` is

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[0]` is 'F'.

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[1]` is

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[1]` is `'r'`.

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[-1]` is

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[-1]` is 's'.

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[3:6]` is

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[3:6]` is 'day'.

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[:3]` is

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[:3]` is 'Fri'.

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[:-1]` is

More on Strings: Indexing & Substrings

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- Strings are made up of individual characters (letters, numbers, etc.)
- Useful to be able to refer to pieces of a string, either an individual location or a “substring” of the string.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|----|----|-----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| F | r | i | d | a | y | s | S | a | ... | S | u | n | d | a | y | s |
| | | | | | | | | | | | | ... | -4 | -3 | -2 | -1 |

- `s[:-1]` is 'FridaysSaturdaysSunday'.
(no trailing 's' at the end)

More on Strings: Splits

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- `split()` divides a string into a list.

More on Strings: Splits

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- `split()` divides a string into a list.
- Cross out the delimiter, and the remaining items are the list.

More on Strings: Splits

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- `split()` divides a string into a list.
- Cross out the delimiter, and the remaining items are the list.

"Friday~~s~~Saturday~~s~~Sunday"

More on Strings: Splits

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- `split()` divides a string into a list.
- Cross out the delimiter, and the remaining items are the list.

```
"FridayXSaturdayXSunday"  
days = ['Friday', 'Saturday', 'Sunday']
```

More on Strings: Splits

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- `split()` divides a string into a list.
- Cross out the delimiter, and the remaining items are the list.

```
"FridayXSaturdayXSunday"  
days = ['Friday', 'Saturday', 'Sunday']
```

- Different delimiters give different lists:

More on Strings: Splits

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- `split()` divides a string into a list.
- Cross out the delimiter, and the remaining items are the list.

```
"FridaysSaturdaysSunday"  
days = ['Friday', 'Saturday', 'Sunday']
```

- Different delimiters give different lists:

```
days = s[:-1].split("day")
```

More on Strings: Splits

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- `split()` divides a string into a list.
- Cross out the delimiter, and the remaining items are the list.

```
"FridaysSaturdaysSunday"  
days = ['Friday', 'Saturday', 'Sunday']
```

- Different delimiters give different lists:

```
days = s[:-1].split("day")  
"FridaysSaturdaysSunday"
```

More on Strings: Splits

```
s = "FridaysSaturdaysSundays"  
days = s[:-1].split("s")
```

- `split()` divides a string into a list.
- Cross out the delimiter, and the remaining items are the list.

```
"FridaysSaturdaysSunday"  
days = ['Friday', 'Saturday', 'Sunday']
```

- Different delimiters give different lists:

```
days = s[:-1].split("day")  
"FridaysSaturdaysSunday"  
days = ['Fri', 'sSatur', 'sSun']
```

Recap

- In Python, we introduced:

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

Recap

- In Python, we introduced:
 - For-loops

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

Recap

- In Python, we introduced:

- ▶ For-loops
- ▶ `range()`

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

Recap

- In Python, we introduced:

- ▶ For-loops
- ▶ `range()`
- ▶ Variables: ints and strings

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

Recap

- In Python, we introduced:

- ▶ For-loops
- ▶ `range()`
- ▶ Variables: ints and strings
- ▶ Some arithmetic

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```


Recap

- In Python, we introduced:

- ▶ For-loops
- ▶ `range()`
- ▶ Variables: ints and strings
- ▶ Some arithmetic
- ▶ String concatenation

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

Recap

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

● In Python, we introduced:

- ▶ For-loops
- ▶ `range()`
- ▶ Variables: ints and strings
- ▶ Some arithmetic
- ▶ String concatenation
- ▶ Functions: `ord()` and `chr()`

Recap

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

- In Python, we introduced:

- ▶ For-loops
- ▶ `range()`
- ▶ Variables: ints and strings
- ▶ Some arithmetic
- ▶ String concatenation
- ▶ Functions: `ord()` and `chr()`
- ▶ String Manipulation

Recap

```
1 #Predict what will be printed:
2 for i in range(4):
3     print('The world turned upside down')
4 for j in [0,1,2,3,4,5]:
5     print(j)
6 for count in range(6):
7     print(count)
8 for color in ['red', 'green', 'blue']:
9     print(color)
10 for i in range(2):
11     for j in range(2):
12         print('Look around,')
13     print('How lucky we are to be alive!')
```

- In Python, we introduced:

- ▶ For-loops
- ▶ `range()`
- ▶ Variables: ints and strings
- ▶ Some arithmetic
- ▶ String concatenation
- ▶ Functions: `ord()` and `chr()`
- ▶ String Manipulation

Weekly Reminders!



Before next lecture, don't forget to:

- Read and work through LAB 2!

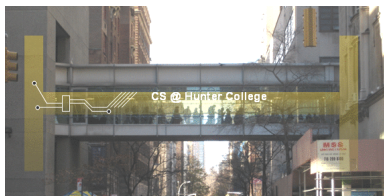
Weekly Reminders!



Before next lecture, don't forget to:

- Read and work through LAB 2!
- Submit this week's 10 programming assignments (programs 1-10)

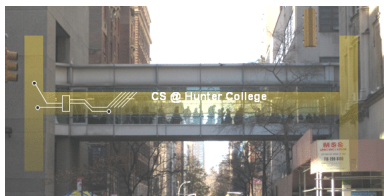
Weekly Reminders!



Before next lecture, don't forget to:

- Read and work through LAB 2!
- Submit this week's 10 programming assignments (programs 1-10)
- If you need help, email cscisummer23@gmail.com with questions or to sign up for Friday online tutoring.

Weekly Reminders!



Before next lecture, don't forget to:

- Read and work through LAB 2!
- Submit this week's 10 programming assignments (programs 1-10)
- If you need help, email cscisummer23@gmail.com with questions or to sign up for Friday online tutoring.