

CS 21: Intro to CS

David Mauskop
mauskop@cs.swarthmore.edu

Why take this course?

- Programming is a powerful skill
- Computer science underlies important technology

This week

- A little unix
- A dash of vim
- A pinch of Python
- Lab 0 due Saturday at midnight

Admin

- Course website: cs.swarthmore.edu/~mauskop/cs21/s17/
- Office hours: Thursday, 2-4pm in SCI 262A
- Email: mauskop@cs.swarthmore.edu
 - Allow one business day for email response
- Ninjas: Nhung, Rye, Zoe
- Attend labs
- Evening ninja sessions
 - Tue 7-9pm, Wed 7-10pm, Fri 7-9pm (all in this room)

Today

- How computers represent data
- Types of data Python makes available
- Python as a calculator
- Python as more than a calculator

Programs Manipulate Data

- Computer data is stored as 0's and 1's, or bits

0's and 1's

10100	A
10101	B
10110	C
10111	D
...	...

0's and 1's in the real world

- Count to 1000 on your hands

0's and 1's in the real world

- Count to 1000 on your hands
 - Each finger is either 'down' or 'up'
 - There are 1024 possible configurations

Sualkuchi silk village







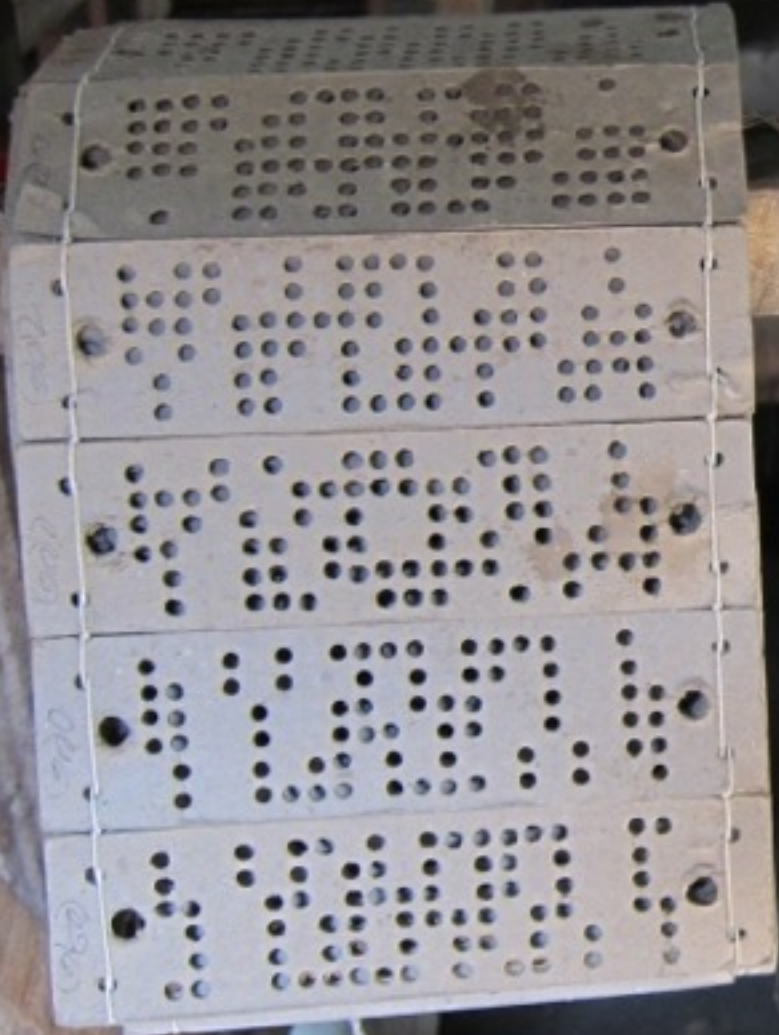








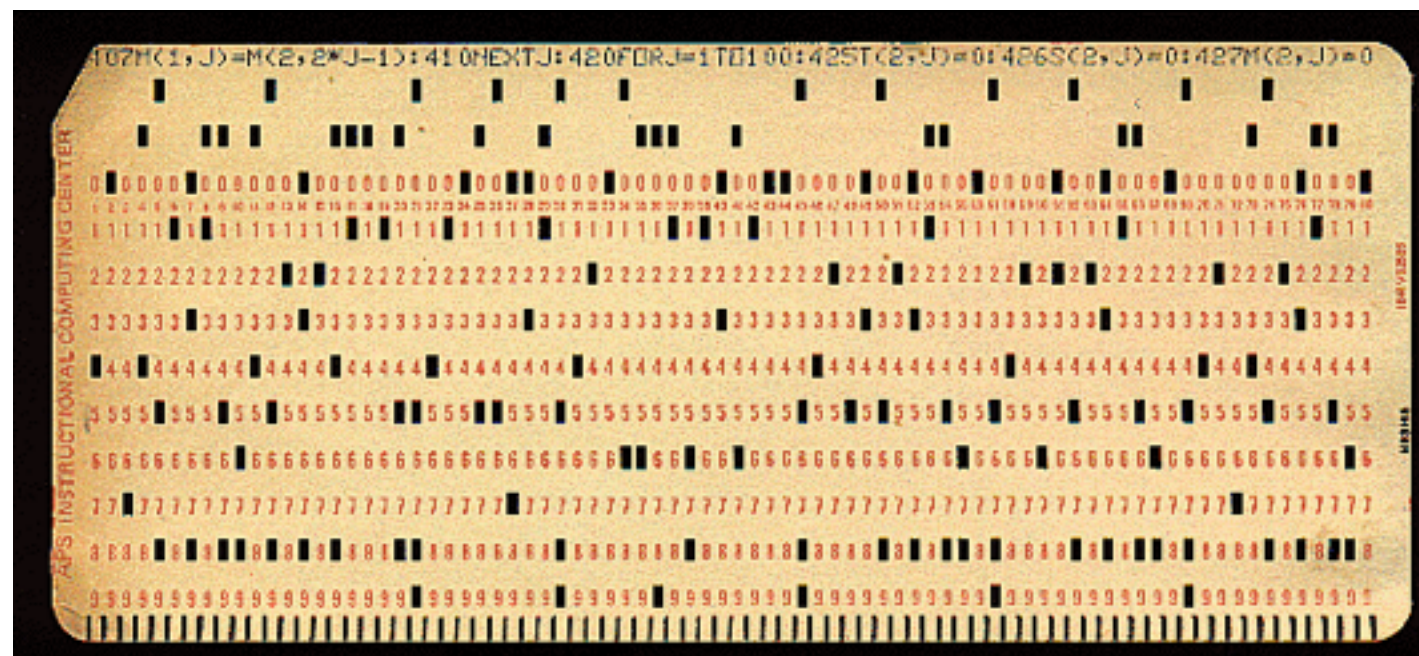
SHAKE
BATHING
NEEDS
MATERIALS



down
Phai
Nurs

Punch card data storage

- 6 rows, 20 columns = 120 potential holes
- $2^{120} =$
1329227995784915872903807060280344576
- Punch cards were used in the early days of computing



Sualkuchi silk village

- Punch cards fed into looms encode different decorative patterns
- A computer's memory is just like these punch cards, except multi-purpose, reusable
- Thinking like a computer scientist doesn't require computers

Python data types

- Three we'll talk about today:
 - integers
 - floating-point numbers
 - strings

Every value has a type

- ints: -3, 4, 1039
- floats: -3.0, 4.006, 3.14
- strings: "hello", 'to be or not to be', "-3", "-3.0"

Python as a calculator

- Operators: +, -, *, /, %
- Different behavior for different types
- Python shell

The + operator

```
>>> 2 + 2
```

```
4
```

```
>>> -3 + 4 + 10
```

```
11
```

```
>>> 5.1 + 3.1
```

```
8.2
```

```
>>> 2 + 2.379
```

```
4.379
```

The + operator

```
>>> "swart" + "more"
```

```
'swarthmore'
```

```
>>> "swarth" + "" + "more"
```

```
'swarthmore'
```

```
>>> 'welcome ' + 'to' + ' ' + 'CS21'
```

```
'welcome to CS21'
```


The - operator

```
>>> 10 - 2
```

```
8
```

```
>>> 5 - 7
```

```
-2
```

```
>>> 5.0 - 3.5
```

```
1.5
```

```
>>> 5 - 3.5
```

```
1.5
```

The * operator

```
>>> 3 * 3
```

```
9
```

```
>>> 2 * 3 * 4
```

```
24
```

```
>>> 3 * 3.0
```

```
9.0
```

```
>>> "panda" * 7
```

```
'pandapandapandapandapandapandapanda'
```

/ and %

```
>>> 10 / 2
```

```
5
```

```
>>> 10 / 2.0
```

```
5.0
```

```
>>> 6.0 / 5.0
```

```
1.2
```

```
>>> 6 / 4
```

```
1
```

```
>>> 6 % 4
```

```
2
```

```
>>> 1 / 2
```

```
0
```

```
>>> 1 % 2
```

```
1
```

Better than a calculator

- Functions
- Complex expressions
- Naming and storing values
- Plus many other capabilities...

First functions

- A function transforms one value into another value, possibly of a different type
- 'len' gives the length of a string as an int
- 'type' gives the type of a value

First functions

```
>>> len('hello')
5
>>> len('')
0
>>> type(12)
<type 'int'>
>>> type(100.0)
<type 'float'>
>>> type("swarthmore")
<type 'str'>
```

First functions - conversion

- The functions 'int', 'float', and 'str' attempt to convert their **arguments**, or inputs, into a value of that type

First functions - conversion

```
>>> float(10)
10.0
>>> float("10.5")
10.5
>>> int(10.43)
10
>>> int("54")
54
>>> int('hello')
ValueError: invalid literal for int()...
>>> str(10)
'10'
>>> str(7.7)
'7.7'
>>> str("i'm already a string")
'i'm already a string'
```


Expressions

- More than simple arithmetic
- An expression is:
 - A value
 - Two or more expressions combined by an operator
 - A function with an expression as input
 - An expression wrapped in parentheses
- Python **reduces** or **evaluates** expressions until they result in a value

Expressions

```
>>> len("hello") + 7
12
>>> type(len("swarth" + "more"))
<type 'int'>
>>> (5 + 7.0) / (9 - 6)
4.0
```

Naming and Storing Values

```
>>> a = 2
>>> a
2
>>> b = a + a + a
>>> b
6
>>> b = b + 1
>>> b
7
>>> c = a * "hi!"
>>> c
'hi!hi!'
>>> lengthOfC = len(c)
>>> lengthOfC
6
```

Naming and Storing Values

- We can **assign** values to named **variables**
- The variable name is a short way of referring to the saved value
- The value can be the result of any expression
- Variable names are themselves expressions
- We can later re-assign a new value to an existing variable

Questions?

See you Friday!

