Problem Solving ...

Building blocks

Excellent Software can be Written in Any Language...

The story of Dave McKay and Paul Salsbury, Accounting and:

- Sequence
- Iteration
- Selection
- Exceptions
- Functions
- Scalars
- Lists
- Dictionaries

McKay and Salsbury Wrote Accounting Software...

Award winning software, in assembly language

Computers are Good at	People are Good at
repetitive tasks	communication
doing exactly what they are told	creativity
data manipulation	empathy
multi-tasking	adapting to new things

McKay and Salsbury did what someone programming in any language would do. They broke accounting down in to things that the computer could do well. Using these building blocks.

Sequence

Python is imperative

Left to right, top to bottom

Input, processing, Output

```
#input

nCelsius = int(input("what is the
temperature in celsius?"))

#processing

nFahrenheit = nCelsius * 1.8 + 32

#output

print("The temperature in
fahrenheit is", nFahrenheit)
```

Iteration

The same thing over and over

For

While

- Note the `:` at the start of an indented block of code

```
#for

for n in range(0,10):

     print("I love Vivas ... please buy some")

#while

n = 0

while n < 10:

     print("I bought some last week ... Did you eat them already?")

     # must increment or infinite loop

     n += 1
```

Selection

Simple decisions

```
if … :

elif …:

else:
```

- Note the `:` at the start of an indented block of code.

```
nGrade = int(input("please enter your grade"))


if nGrade < 50:
    print("you failed")
elif nGrade >= 90:
    print("A+")
else:
    print("you passed")
```

Exceptions

Say the user enters `asdf`

Feel good case

Exceptional case

```
try:
      nGrade = int(input("please enter your
grade"))
      if nGrade < 50:
      print("you failed")
      elif nGrade >= 90:
      print("A+")
      else:
      print("you passed")
except:
      print("please enter a grade between 0 and
100")
```

Functions

Named and re-usable lumps of code

Abstraction for Salsbury, McKay (and team)

```
def celsius2fahrenheit(nCelsius):

      return nCelsius * 1.8 + 32


#input

nCelsiusInput = int(input("what is the
temperature in celsius?"))

#processing

nFahrenheit =
celsius2fahrenheit(nCelsiusInput)

#output

print("The temperature in fahrenheit is",
nFahrenheit)
```

Scalars

We have already used

From grade 9 math???

Let x = 7

I try to use system hungarian notation

Simonyi (a Hungarian) made this popular co-writing the first version of MS-Word

Python is dynamically typed

Prefixes remind me of the type

n	number
a	List (array)
dict	Dictionary
s	String
o	Object

Lists

Programmers start counting at 0

In math we called them arrays

The range function we used in the iteration example returns a list

[0,1,2,3,4,5,6,7,8,9]

```
def day2dayOfWeek(nDay):

	aDays = ["Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday",

	"Friday", "Saturday"]

	return aDays[nDay - 1]

try:

	nDay = int(input("Enter a day of the week ... 1
for Sunday: "))

	sDayOfWeek = day2dayOfWeek(nDay)

	print("The", nDay, "of the week is", sDayOfWeek)

except:

	print("please enter a day from 1-7")
```

Dictionaries

Associative Arrays

Likely not available to Salsbury and McKay or Simonyi

Indices are objects rather than 0,1,2,3

Often strings

```
dictCapitals = {"ON":"Toronto","MB":"Winnipeg","BC":"Victoria"}

def getCapital(sProv):

        global dictCapitals

        return dictCapitals[sProv]

try:

        sAbbreviation = input("Enter a short form for a province:
")

        sCapital = getCapital(sAbbreviation)

        print("The capital of", sAbbreviation, "is", sCapital)

except:

        # there is a lot going on here!

        print("Please enter a province from",
list(dictCapitals.keys()))
```

Programming Decomposes a Problem ...

Into things a computer does well

For instance multiplying and dividing

Brains are good at creativity and responding to new things

The trick when learning to code is to creatively and logically give the computer what it needs to solve the problem at hand

Next time objects...