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
- Functions
- Scalars
- Arrays
- Strings
- Associative Arrays

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

```
import readlineSync from
'readline-sync';

//input

let nCelsius =
readlineSync.question("what is the
temperature in celsius? ");

//processing

let nFahrenheit = nCelsius * 1.8 + 32;

//output

console.log(`The temperature in
fahrenheit is ${nFahrenheit}`);
```

Iteration

The same thing over and over

For

While

- Note the `{` at the start and `}` at the end of an indented block of code

```
// n counting down
for(let n = 10; n > 0; n = n - 1){
     console.log("I love Vivas ... please buy some");
}
// n counting up
let n = 0;
while(n < 10){
     console.log("I bought some last week ... Did you eat them already?");
     // if you don't add to n it will always be less than 10
     n = n + 1;
}
```

Selection

Simple decisions

if … {

else if …{

else{

- Note the `{` at the start and `}` at the end of an indented block of code.

```
import readlineSync from 'readline-sync';

//input

let nGrade = readlineSync.question("Enter your grade out of 100? ");

//select among letter grades

if(nGrade < 50){

      console.log("you failed");

}else if(nGrade >= 90){

      console.log("A+");

}else{

      console.log("you passed");

}
```

Functions

Named and re-usable lumps of code

Abstraction for Salsbury, McKay (and team)

```
import readlineSync from 'readline-sync';

let celsius2fahrenheit = (nCelsius)=>{

       return nCelsius * 1.8 + 32;

}

//input

let nCelsiusInput = readlineSync.question("what is the temperature in celsius? ");

//processing

let nFahrenheit = celsius2fahrenheit(nCelsiusInput);

//output

console.log(`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

Javascript is loosely typed

Prefixes remind me of the type

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

Arrays

Programmers start counting at 0

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

```
import readlineSync from 'readline-sync';

let day2dayOfWeek = (nDay) => {

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

       "Friday", "Saturday"]

       return aDays[nDay - 1];

}

//input

let nDayOfWeek = readlineSync.question("Enter a day of
the week ... 1 for Sunday: ")

//processing

let sDayOfWeek = day2dayOfWeek(nDayOfWeek);

//output

console.log(`The day of the week is ${sDayOfWeek}`);
```

Strings

String manipulation is a special case of list processing.

Some operators also apply

Experiment with +><

+	Concatenates strings
><	Compares strings in lexical order (like dictionary)

```
import readlineSync from 'readline-sync';
let countUpperCase = (sInput) =>{
          let nUpper = 0;
          for(let n = 0; n < sInput.length; n = n+ 1){
          if(sInput[n] != sInput[n].toLowerCase()){
          nUpper = nUpper + 1;
          }
          }
          return nUpper;
}
//input
let sInput = readlineSync.question("Enter a string with both upper and lower case: ");
//processing
let nUpper = countUpperCase(sInput);
//output
console.log(`Your string has ${nUpper} uppercase letters`)
```

Associative Arrays

Foundation of objects

Likely not available to Salsbury and McKay or Simonyi

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

Indices are often strings

```
import readlineSync from 'readline-sync';

let getCapital = (sProv)=>{

      let oCapitals =
{"ON":"Toronto","MB":"Winnipeg","BC":"Victoria"}

      return oCapitals[sProv];

}

//input

let sProvShortForm = readlineSync.question("Enter a
province abbreviation: ");

//processing

let sCapital = getCapital(sProvShortForm);

//output

console.log(`The capital of ${sProvShortForm} is
${sCapital}`);
```

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...