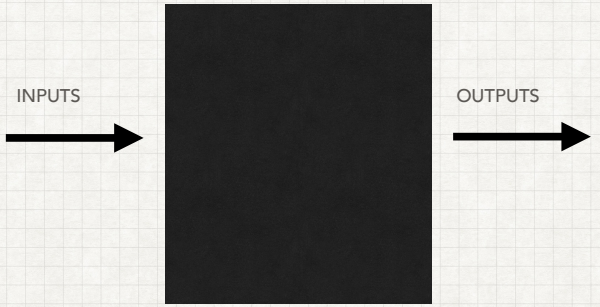
JAVASCRIPT

TEXT BASED 'CODING'

- use a text editor to "write" code
- an interpreter takes our code and translates it into computer speak — lower level language designed for computers, not humans

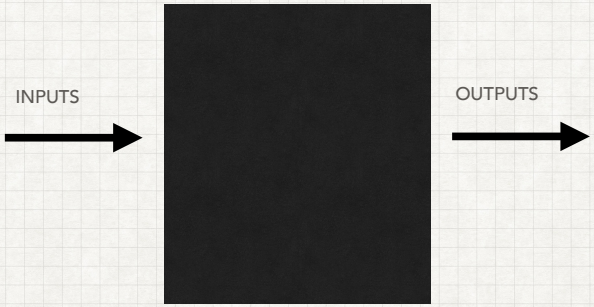
17

Computer Science:



18

PROGRAMMING: making the black box work



19

PROGRAMMING

CREATING THE BLACK BOX

- Start with a problem
- Know the constraints: the input(s), the expected output(s), etc
- Develop a solution

20

PROGRAMMING

3 STEPS

1. Understand the problem, constraints, goals
2. Create the logic to solve it
   - tools: pictures, research, pseudo code
   - modeling the data
3. Write: translate into computer language (e.g. javascript)
   - tools: research, reading, learning apis
   - document logic

21

PROGRAMMING

SOLVING THE BLACK BOX

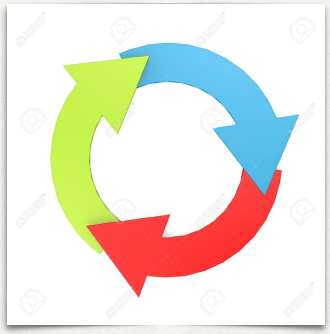
- programming is taking our logic and translating into javascript
- pseudo coding (in between thoughts and javascript)
  - draw pictures,
  - write 'code-like' statements
  - less frequent as we get more experience
  - map into javascript statements/constructs

22

PROGRAMMING

ENDLESS "LOOP"

Write/Document, Run/Test, Debug



23

EXAMPLE:

determine if a number is even or odd

24

STEP 1: PROBLEM

DETERMINE IF A NUMBER IS EVEN

- What do we need to know?
  - what "numbers" will be used (the domain)
  - what should our result be? (true/false, 'yes/no', ??)
  - how will others be using this?

STEP 2: LOGIC

HOW CAN WE DETERMINE IF A NUMBER IS EVEN?

```
is 12345 an even number ?

How do we decide if a number is even or
not ?
```


STEP 2 (CONT): PSUEDO CODE

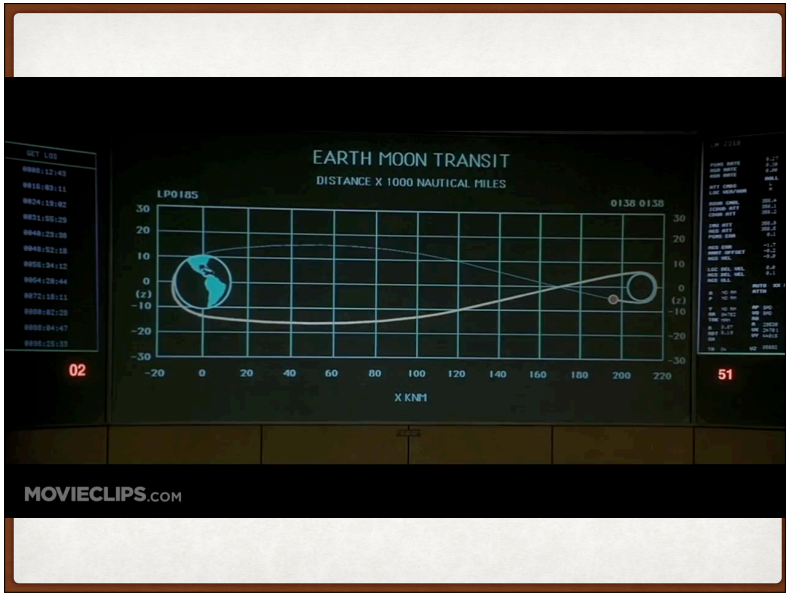
HOW CAN WE DETERMINE IF A NUMBER IS EVEN?

```
write code here

e.g. is 12345 an even number ?
```

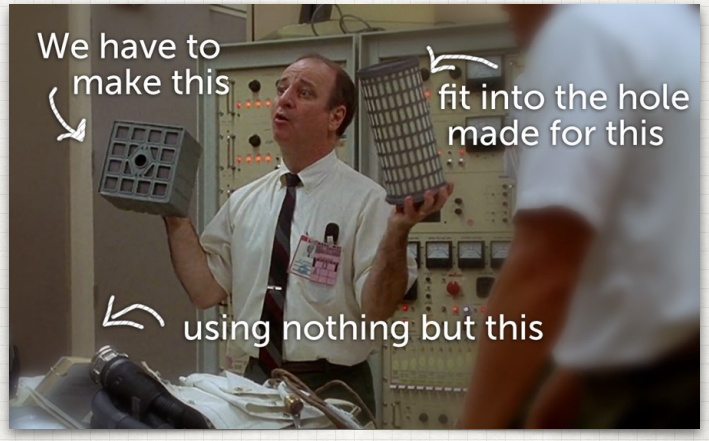

STEP 3: WRITE/TRANSLATE

HOW CAN WE DETERMINE IF A NUMBER IS EVEN?

```
// write code here

// is 12345 an even number ?

// NEED to know what tools we have

// re-think our solution ?
```



29



30

PROGRAMMING CONSTRUCTS

CODING STRUCTURE TO KEEP US SANE

- **Variables** (remember a value)
  - data, value & type
- **Branching** (conditional code)
  - boolean expressions
- **Loops** (repeat, iterate, enumerate)
  - boolean expressions
- **Code Organization**
  - functions, objects,

31

repl.it
(demo time)

32

EXAMPLE:

reuse even/odd functionality

33

function anatomy

```
function isEven(num) {


}
```

34

function anatomy

```
function isEven(num) {


}
```

35

function anatomy

```
function isEven(num) {


}
```

36

function anatomy

```
function isEven(num) {


}
```

37

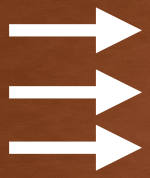
function anatomy

```
function isEven(num) {


}
```

38

function anatomy

```
function isEven(num) {


}
```



39

function anatomy

```
function isEven(num) {

  return true;
  // return false;
  // return 0;
  // return "Yes";
  // return ;
}
```

40

function anatomy

```
function isEven(num) {
  //pseudo code
  if (num is even) return true
  otherwise return false

}
```

41

CONDITIONAL BRANCHING

42

```
if (boolean_expression) {


}
```

43

```
if (boolean_expression) {


}
```

44

```
if (boolean_expression) {


}
```

45

BOOLEAN EXPRESSIONS

EVALUATE TO TRUE OR FALSE

	INVERSE
X == Y	X != Y
X > Y	X <= Y
X < Y	X >= Y

46

BOOLEAN EXPRESSIONS

EVALUATE TO TRUE OR FALSE

```
AND:   A && B
OR:    A || B
NOT: !A


A && !B || (C && D)
```

47

```
if (boolean_expression) {


}
```

48

```
if (boolean expression) {

  // execute if true

} else {

  // execute if false
}
```

49

```
if (exp1) {
  // execute if true
}else if (exp2) {
  // execute if true
}else {
  // execute both false
}
```

50

```
function isEven(num) {

  if (num%2 == 0) {
    return true;
  }
  else {
    return false;
  }
}
```

51

```
function isEven(num) {

  if (num%2 == 0) {
    return true;
  }
  return false;
}
```

52

```
function isEven(num) {

  return (num%2 == 0);

}
```

53

EXAMPLE:

in a set of numbers, how many are even?

54

	A	B	C	D	E
1	23	42	4	15	6

```
isEven(23);
isEven(42);
isEven(4);
isEven(15);
isEven(6);
```

55

```
if (isEven(23)) {
  // increment counter
}
if (isEven(42)) {
  // increment counter
}
. . .
```

56

```
if (isEven(23)) {
  // increment counter
}
if (isEven(42)) {
  // increment counter
}
. . .
```

57

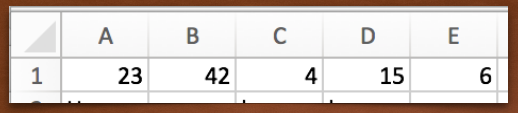
```
var evenCount = 0;
if (isEven(23)) {
  // increment counter
  evenCount = evenCount + 1;
}
if (isEven(42)) {
  evenCount += 1;
}
if (isEven(4)) {
  evenCount++;
}
```

58

ARRAYS

BETTER WAY TO HOLD DATA

59



A1:E1

[23, 42, 4, 15, 6]

60