

Custom Functions

Custom Functions

- We've done this before (button functions)
- Only new concept: Passing arguments (data) we actually want to use.
- Why?
 - To be lazy, to make changes easier and more widespread, all the typical stuff
- Syntax (should be familiar)


```
(variableName:variableType, nextName:nextType)
```
- Example:


```
(evt:MouseEvent)
```

Once you have the arguments

- Use them inside function just like a variable.
- For example:


```
function showScore(nCorrect:int, nWrong:int) {
  var score:Number;
  score = nCorrect / (nCorrect + nWrong);
  trace(score);
}
```
- Function call:


```
showScore(25, 75);
```

Returning data

- Functions can give you a “result”
- Needs a type
- Uses keyword return

■ For example:

```
function showScore(nCorrect:int, nWrong:int):String {  
    var score:Number;  
    score = nCorrect / (nCorrect + nWrong);  
    return String(score);  
}
```

■ Function call:

```
resultsText.text = showScore(25, 75);
```

Variable “scope”

- Passed to a function or created inside a function
 - Short time to live
- Functions can also “pull” variables from outside the function.
- Order: check inside first, then outside

Functions and pathing

- It's not about where you make the function call from
- It's about where the function “lives”

Example (Kendra Hall)

- Functions to do custom start drag
- Functions to do custom stop drag
- A function to update a progress gauge


