Announcement

No:

- Midterms
- Final

Yes:

- Labs
- Quizzes
- Projects
- Quick Writes

Project 2B

- Project 2B and its “quiz” are linked on our online calendar
  - “Quiz” is
    - Open book
    - Save and resume
    - Retake as often as you want to improve your score
    - Be sure to submit it before the quiz closes!

Quiz and Quick Write

- Next week
  - Review the questions at the end of these chapters:
    - *Fluency* chapters 18 and 20
    - *QuickStart* chapters 1 and 2
- All JavaScript!

- Topics will include:
  - Variables
  - Values & data types
  - Assignment statements
  - Rules for identifiers
  - Conditionals
  - Loops
  - Arrays
  - Functions
  - Curly brackets

Thinking like a computer thinking like a human being....

CONTROL FLOW

Fitting it together...

- An algorithm is....
  - Write one sentence on a strip of paper

Fitting it together…

- An algorithm is….
  - A set of directions
  - Listed sequentially
    - Start at beginning
    - Continue
    - Until you reach the end

Leaving Lecture algorithm

1. Start in your seat at Mary Gates Hall 389
2. Pack up your stuff
3. Pick it up
4. Stand up
5. Walk to end of aisle
6. Walk down steps until you reach bottom of steps
7. Turn left
8. Walk through doors

Control Flow

- Control flow is the sequence through the code
- What we just looked at was ***sequential*** flow
  - Start at step 1 continue through step 8
- Now we'll look at others....

Fitting it together...

- An algorithm is....
  - A set of directions
  - Listed sequentially
    - Start at beginning
    - Continue
    - Until you reach the end
  - Conditionals, or tests, change the control flow

Leaving Lecture algorithm

1. Start in your seat at Mary Gates Hall 389
2. Pack up your stuff
3. Pick it up
4. Stand up
5. Walk to end of aisle
6. Walk down steps until you reach bottom of steps
7. Turn left
8. Walk through doors

Fitting it together…

- An algorithm is….
  - A set of directions
  - Listed sequentially
    - Start at beginning
    - Continue
    - Until you reach the end
  - Change the control flow with
    - Conditionals, or tests

Leaving Lecture algorithm

1. Start in your seat at Mary Gates Hall 389
2. Pack up your stuff
3. Pick it up
4. Stand up
5. Walk to end of aisle
6. Walk down steps until you reach bottom of steps
7. Turn left
8. Test: Is door open?
   1. Yes: Walk through doors
   2. No: Open door, then walk through

Fitting it together…

- An algorithm is….
  - A set of directions
  - Listed sequentially
    - Start at beginning
    - Continue
    - Until you reach the end
  - Change the control flow with
    - Conditionals, or tests
    - Iteration, or loops

Leaving Lecture algorithm

1. Start in your seat at Mary Gates Hall 389
2. Pack up your stuff
3. Pick it up
4. Stand up
5. Walk to end of aisle
6. Loop:
   1. Walk down 1 step at a time until you reach the bottom
7. Turn left
8. Test: Is door open?
   1. Yes: Walk through doors
   2. No: Open door, then walk through

Fitting it together…

- An algorithm is….
  - A set of directions
  - Listed sequentially
    - Start at beginning
    - Continue
    - Until you reach the end
  - Change the control flow with
    - Conditionals, or tests
    - Iteration, or loops

Leaving Lecture algorithm

1. Start in your seat at Mary Gates Hall 389
2. Test: Any stuff out?
   1. True: Pack up your stuff
3. Pick it up
4. Stand up
5. Walk to end of aisle
6. Loop:
   1. Walk down 1 step at a time until you reach the bottom
7. Turn left
8. Test: Is door open?
   1. True: Walk through doors
   2. False: Open door, then walk through

Moving the data on the form…

MORE FORMS

Events Cause Processing

After drawing a page, browsers sit idle waiting for something to happen ... when we give input, it cause *events*

- Processing the input is the task of an *event handler*
  - Event types
    - onClick
    - onChange
    - onMouseOver

In the <input ...> tag, an event handler gives the processing needed for the task using JavaScript

Demonstration

- Smileys...

Asian Emoticons

(^_^)	Laughing	(???)/	Joyful
(> <)>	Troubled	(???;)	Surprised
(^_^;)	Troubled	(#^.^#)	Shy
(ToT)	Crying	(*´?`*)	Infatuation
m(_ _)m	Apologising	(??;)	Worried
(^^;)	Shy	(*^?^*)	Joyful
(???)	Grinning	(^?^)	Laughing

Rightside up

Emoticons = Emotional Icons

:-)	Smile or Happy
:-(	Frown or Sad
;-)	Winking
:-D	Laughter
:-C	Very, very sad
D-:	Annoyed, shocked or scared
:-p	“Raspberry" or 'tongue in cheek’
:-S	Confused
:-/	Doubtful or confused
:-|	Blank
O:O_O	Surprised or shocked

Observe Actions

Emoticons

: -) ; -) : -(=8-O

Adding a smile to

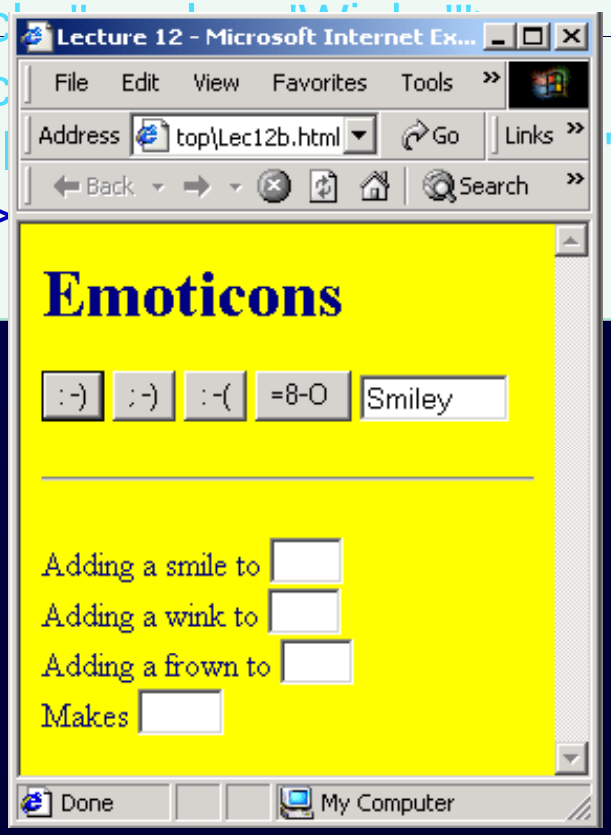
Adding a wink to

Adding a frown to

Makes

‘onClick’ Event for Buttons

```
<h1>Emoticons  </h1>
  <input type="button" value=" : -) " onClick="x.value='Smiley'">
  <input type="button" value=" ; -) " onClic
  <input type="button" value=" : -( " onClic
  <input type="button" value="=8-O " onCl
  <input type="text" name=x  size=8><br>
     ...
```

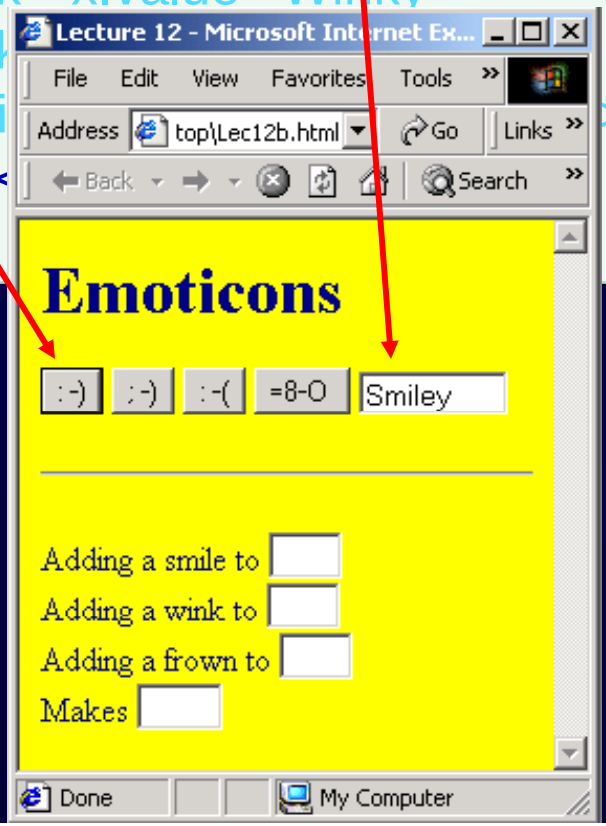


* Event handlers say what to do if event happens ... “put ‘Smiley’ in the output textbox”

Event handlers = mini programs

‘onClick’ for Buttons

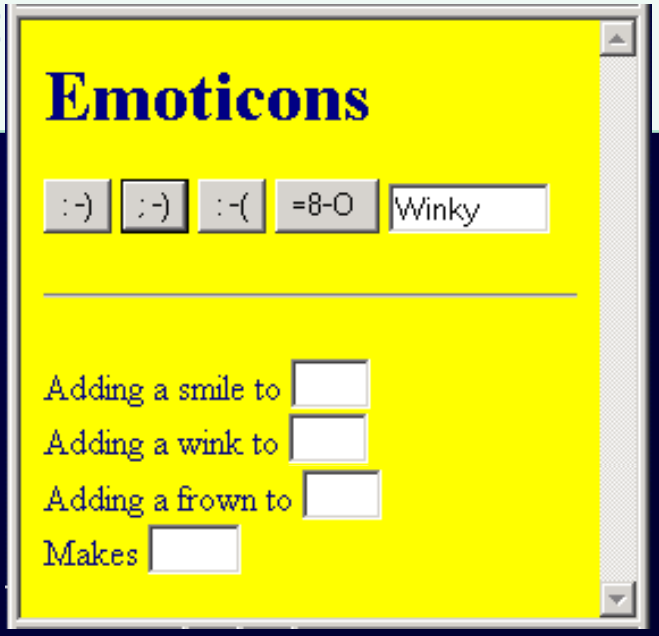
```
<h1>Emoticons  </h1>
  <input type="button" value=" : -) " onClick="x.value='Smiley' ">
  <input type="button" value=" ; -) " onClick="x.value='Winky'">
  <input type="button" value=" : -( " onClick
  <input type="button" value="=8-O " onCli
  <input type="text" name=x  size=8><br><
     ...
```



* Notice …
  - ‘**`onClick`**’ event does the task: places ‘Smiley’ in the output textbox

x.value

```
<h1>Emoticons  </h1>
  <input type="button" value=" : -) " onClick="x.value='Smiley'">
  <input type="button" value=" ; -) " onClick="x.value='Winky'">
  <input type="button" value=" : -( " onClick="x.value='Frowny'">
  <input type="button" value="=8-O " onClick="x.value='Omagosh!'">
  <input type="text" name="x" size="8"><
     ...
```

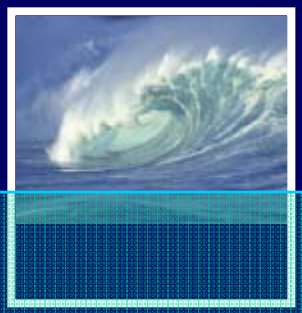


* Notice ...
  - the **value** of a textbox is the contents of the textbox **x.value**

textbox name

Getting the data to the functions...

PASSING VALUES TO FUNCTIONS



Functions

Head

```
function addIt(x, y)
{
    var x, y;
    return x+y;
}
```

Body

```
Result = addIt(6, 7);
```