

FORMS

To create more professional programs, we can use forms for input and output. See the GuessANumber - week 14 ppt example.html file for an explanation. Open the file in a browser to see what the user interface looks like.

We will use two types of inputs/outputs: buttons and text boxes.

1. Buttons. When the user clicks on them, some kind of action happens.

For instance

```
<input type="button" id="startButton" VALUE = "Start New Game" onclick=startGame()>
```

The ID (here: “startButton”) is a unique identifying name we give to the input. Like a variable name. VALUE is assigned a string that will be written on the button in the webpage. Here it is “Start New Game”.

Finally, we see the onclick action. In this case, it is a call to the function startGame(). The parentheses are empty, so no parameters are passed to the function.

The other button code is very similar:

```
<input type="button" id="guessButton" VALUE = "Guess" onclick=makeGuess()>
```

Here the id is "guessButton", and the action on clicking is to call the function makeGuess().

2. Text. This creates a textbox where the user can enter data (e.g. a name, or a number). Similar to the prompt statement. BUT, to make it confusing, the textbox can also be used to write output.

For instance

```
<input type="text" id="guess">
```

If you want a larger text box (e.g. 55 characters) then you can specify that:

```
<input type="text" id="guessResult" size="55">
```

Whenever a user clicks on the Guess button, whatever value was entered in the textbox after “What is your guess?” gets assigned to the input with the id “guess”. At the same time, the function makeGuess() on line 19 is called.

Line 21 uses a Javascript function that is new to us, “document.getElementById” It looks like this:

```
numberGuessed = document.getElementById("guess").value
```

It takes the value from the input with the id “guess” and assigns its value to the numberGuessed variable. In other words whatever was in the textbox is assigned to numberGuessed.

Now for the confusing part: using an *input* textbox for *output*.

On line 26, the value of the guessResult input is set to ‘Your guess was too low’. Before we used alert statements to write results. On line 26 (or 29 or 32), we assign a string to guessResult, and it is immediately displayed in the textbox within the form.

The button from line 38 is used to reset the game. When the user clicks the button, the startGame() function is called on line 12. This clears any text in the “guess” and “guessResult” textboxes.