

# CSC 337

## Widget Purchase

First Name

Last Name

Phone

City

State

Zip

Size

- ☐ Small \$2.00
- ☐ Medium \$2.65
- ☐ Large \$2.99

Quantity (1..5)

Continue

## HTML Input Types

Rick Mercer

# There are many input types

- What we have seen three: input, password, button
- Here is a review with an attribute (or 2) each
- means 0 to 9 characters can be enter
- size means make the input 9 characters wide on the page

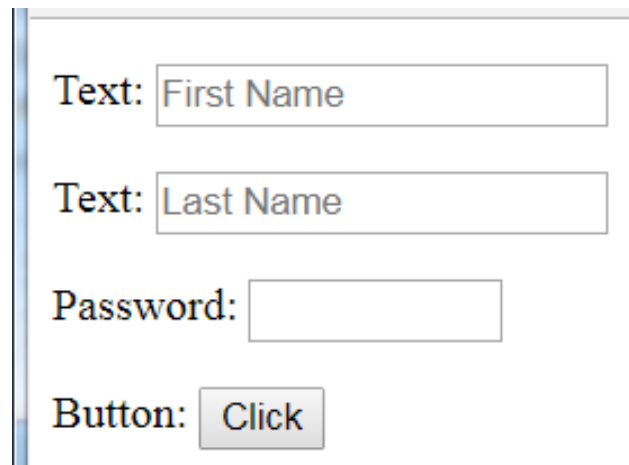
```
<!-- text is the default -->
```

```
<p>Text: <input placeholder="First Name"> </p>
```

```
<p>Text: <input type="text" placeholder="Last Name"> </p>
```

```
<p>Password: <input type="password" maxlength="9" size="9"></p>
```

```
<p>Button: <input type="button" value="Click"> </p>
```



Text:

Text:

Password:

Button:

# Many lines of input, an HTML tag: `<textarea>`

- `textarea`: multi-line text input area (it's an HTML tag)
- `rows="6"` specify height as 6 rows (will score if needed)
- `cols="50"` specify width as 50 characters
- `placeholder="string"` places the string into

`<body>`

`<p>What is your least favorite thing about Google Sites?</p>`

`<textarea rows="6" cols="50" placeholder="Add comments here ..."></textarea>`

`</body>`

What is your least favorite thing about Google Sites?

Add comments here ...

# `<input type="checkbox">`

- Yes/no choices that can be checked and unchecked
- 0, 1, or many checkboxes can be checked at same time
- Can use `checked="checked"` attribute in HTML to initially check a box or two

```
<body>
```

```
<input type="checkbox"> Lettuce
```

```
<input type="checkbox" checked="checked"> Tomato
```

```
<input type="checkbox" checked="checked"> Pickles
```

```
</body>
```

☐ Lettuce ☒ Tomato ☒ Pickles

# `<input type="radio" >`

- Sets of mutually exclusive choices (inline)
- Can group by name attribute so only radio button one can be checked at a time

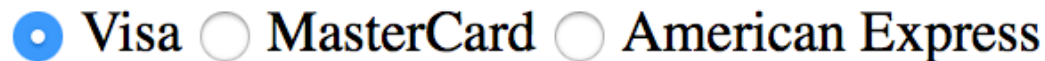
`<body>`

`<input type="radio" name="cc" value="visa" checked="checked">` Visa

`<input type="radio" name="cc" value="mastercard">` MasterCard

`<input type="radio" name="cc" value="amex">` American Express

`</body>`



☒ Visa ☐ MasterCard ☐ American Express

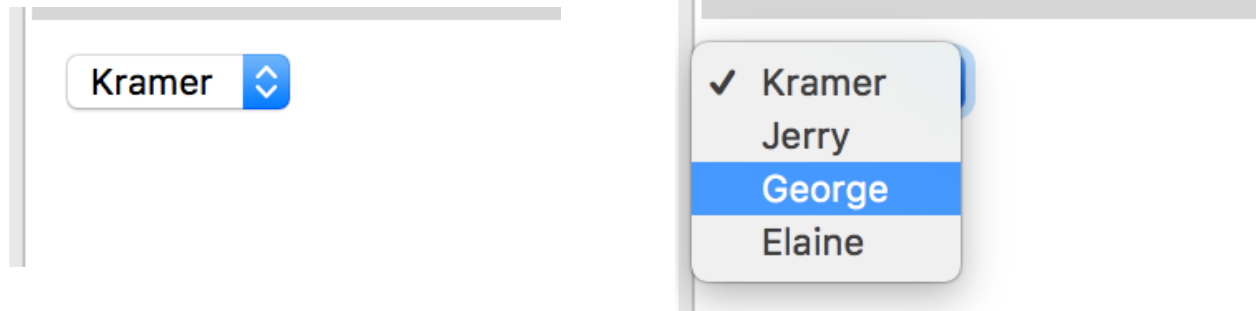
# Drop-down list with `<select>` `<option>`

- Menus of choices that collapse and expand
- `<option>` element represents each choice
- `selected` attribute sets the one initially chosen

`<body>`

```
<select>
  <option selected>Kramer</option>
  <option>Jerry</option>
  <option>George</option>
  <option>Elaine</option>
</select>
```

`</body>`



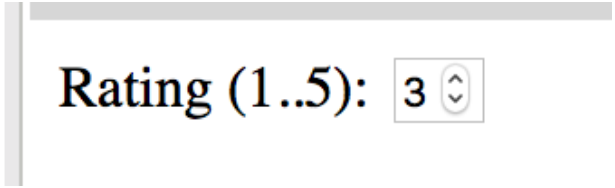
# `<input type="number">`

- For entering integers with a increase/decrease tool on the right
- Not available in all browsers
- Can add optional attributes `min` and `max`

`<body>`

Rating (1..5): `<input type="number" min="1" max="5">`

`</body>`

A screenshot of a web form. It features a label "Rating (1..5):" followed by a number input field. The input field contains the number "3" and has small up and down arrow buttons on its right side, indicating it is a spinner control. The entire form is enclosed in a light gray border.

Rating (1..5): 3

# `<input type="date">` with an event handler

*does not work in Safari, Eclipse, or IE*

```
<body>
```

Select departure date

```
<p><input type="date" id="selectedDate"></p>
```

```
<p><input type="button" value="Show selected date"
onclick="f()"></p>
```

```
<script>
```

```
element = document.getElementById("selectedDate");
```

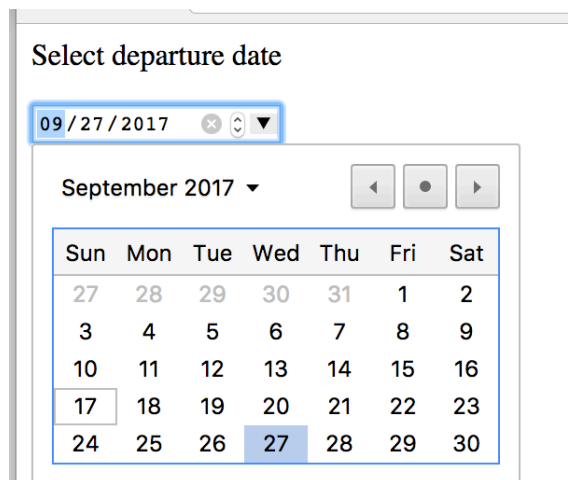
```
function f() {
```

```
    alert('You selected ' + element.value);
```

```
}
```

```
</script>
```

```
</body>
```



Select departure date

09/27/2017

September 2017

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30



# The color input type with an event handler and two style changes

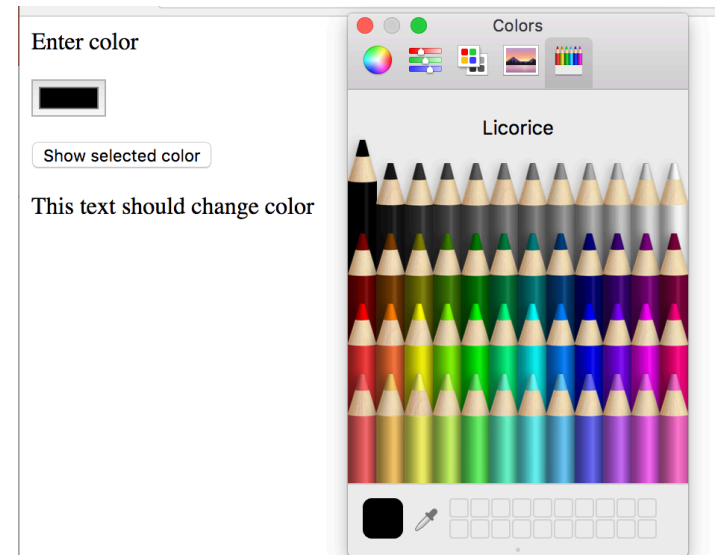
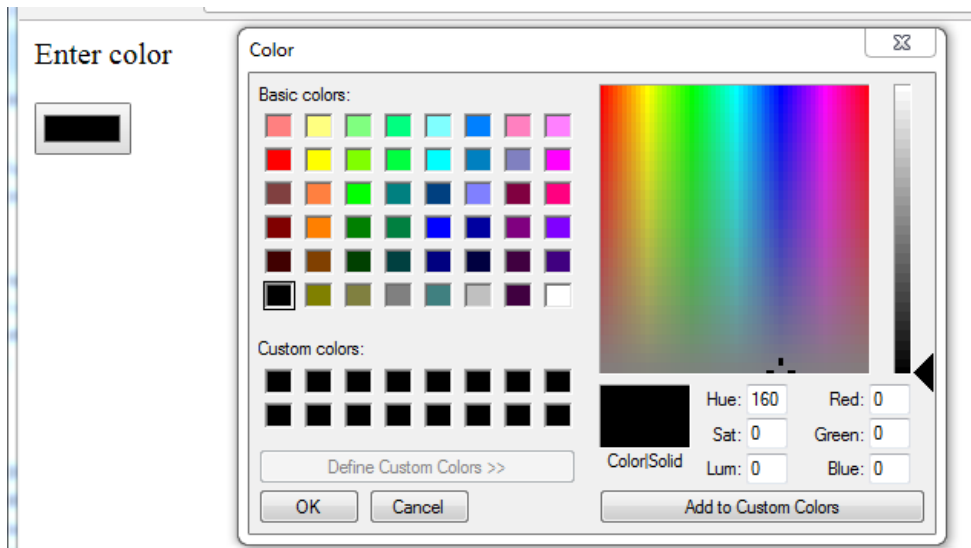
- Input type="color" generates a button
- When clicked, it pulls up a color chooser
- The chosen color is available as the element's **value**

```
<body>
```

```
  Enter color
```

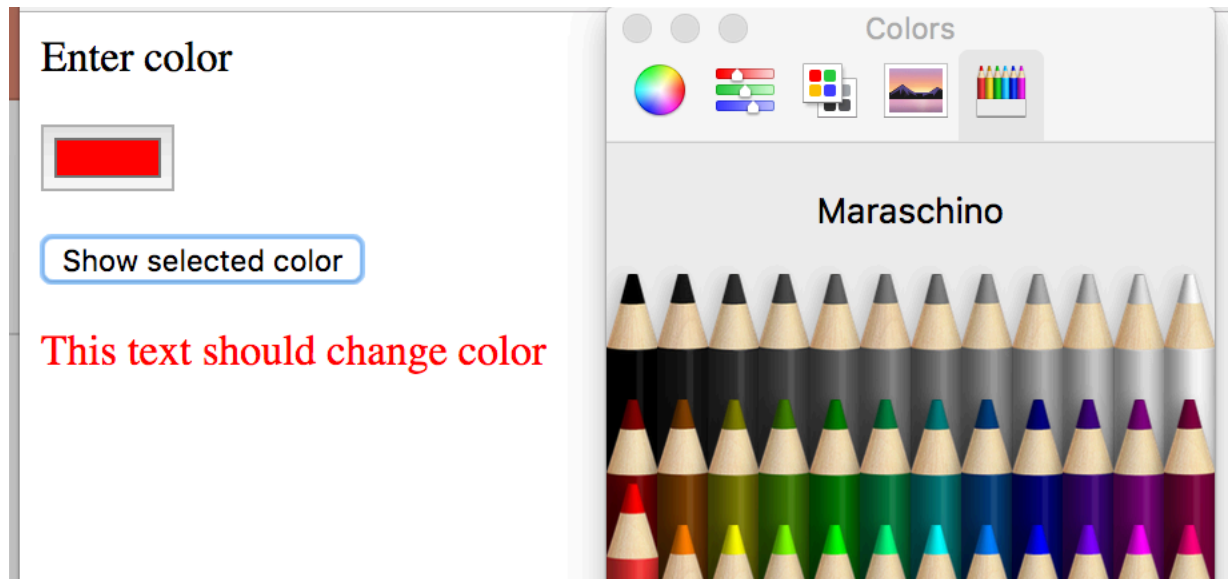
```
  <p><input type="color" id="colorChosen"></p>
```

```
</body>
```



# ICA

- Write an event driven program such that when the button is clicked, the text below the button changes to the selected color





The HTML `<form>`

# The HTML form

- The HTML `<form>` is used to collect user input
- Forms have many HTML input types to collect data
- Typically has one `<input type="submit">`
  - when this is clicked, the entered data can be sent to a server or call a JavaScript function
- General form, attributes could be `action` or `onsubmit`

`<form attributes >`

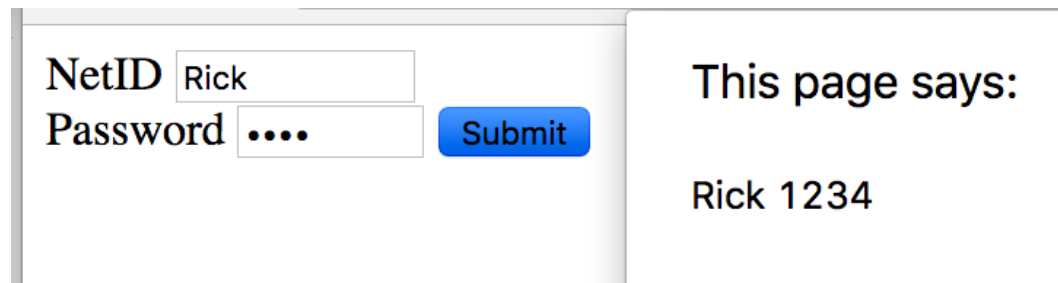
*...input types...*

`</form>`

# Preview of an HTML form

```
<body>
<form onsubmit="jsFunction()">
  NetID <input type="text" id="NetID"> <br>
  Password <input type="password" id="Password">
    <input type="submit">
</form>

<script>
var netIDelement = document.getElementById("NetID");
var passwordElement = document.getElementById("Password");
function jsFunction() {
  alert(netIDelement.value + " " + passwordElement.value);
}
</script>
```



The screenshot shows a web browser window. On the left, there is a form with two input fields: 'NetID' containing the text 'Rick' and 'Password' containing four dots. To the right of these fields is a blue 'Submit' button. On the right side of the browser window, there is a message that says 'This page says: Rick 1234'.

# Using an HTML Form

- A user visits a web page containing a form
- The browser displays the HTML form
  - Usually with two or more input type elements
- The user fills in the form and submits input
  - input can be validated in several ways
- The browser sends the submitted form data to a web server or a JavaScript function
- A form processor script runs on the web server or the JavaScript function executes

# Many things can go into a <form>

- Form elements include, but are not limited to
  - buttons, checkboxes, text fields, radio buttons, lists, ...
  - any HTML element like a <br>

<form>

Enter something <input type="text">

<br>

<input type="checkbox" name="major" checked="checked">

Computer Science

<input type="checkbox" name="ECE">

ECE

</form>

Enter something



Computer Science



ECE

## I can't resist ...

- After the test, we will use forms to gather data to be sent to the server using `action="url"`
- When form has been filled out and submitted, its data will be sent to the action's URL

```
<form action="http://www.google.com/search">  
  <div>  
    Let's search Google: <input name="q">  
    <input type="submit">  
  </div>  
</form>
```



# Prepare to actually submit a form

```
<form>
  <input type="text" id="userID"> Account Name<br>
  <input type="password" id="password"> Password<br> <br>

  <fieldset> <!-- Use fieldsets like a <div> to group things -->
    <legend>Section</legend> <!-- legend also needed in lab -->
    1 <input type="radio" name="section" value="1"> 11:00am<br>
    2 <input type="radio" name="section" value="2"> 1:00pm<br>
    3 <input type="radio" name="section" value="3"> 3:00pm
  </fieldset>
</form>
```

peanut Account Name  
..... Password

## Section

- 1 ☐ 11:00am
- 2 ☒ 1:00pm
- 3 ☐ 3:00pm

## Then make the form 'submittable'

- Define a button for submitting the form to a form-handler, usually goes at the bottom

```
<input type="submit" value="Submit">
```

- The form-handler is typically a server page with a script for processing input data (like the Google url)
- The form-handler can also be a JavaScript function using the event attribute `onsubmit`

```
<form onsubmit="someJSfunction">
```

# Show the Account Name entered

```
<form onsubmit="getTheData()">
  <input type="text" id="userID"> Account Name<br>
  <input type="password" id="password"> Password<br> <br>
  <fieldset> <!-- fieldsets group things like a <div> -->
  <legend>Section</legend> <!-- legend also needed in lab -->
  1 <input type="radio" name="section" value="1"> 11:00am<br>
  2 <input type="radio" name="section" value="2"> 1:00pm<br>
  3 <input type="radio" name="section" value="3"> 3:00pm
</fieldset> <br>
<input type="submit"
      value="Submit">
</form>
<script>
function getTheData() {
  alert("Account: " +
    document.getElementById("userID").value);
}
</script>
```

Guber	Account Name
	Password

Section

1

☐

11:00am

2

☐

1:00pm

3

☐

3:00pm

This page says:

Account: Guber

Submit

# Get the checked radio button

- Can use `document.forms`, an array, and a loop
- `document.forms.length` returns # of forms on a page
- `document.forms[0]` returns the first form

```
<script>
```

```
function getTheData() {  
    var sections = document.forms[0];  
    var sectionChecked = 'not yet';  
    for (var i = 0; i < sections.length; i++) {  
        if (sections[i].checked)  
            sectionChecked = sections[i].value;  
    }  
    alert("You checked section " + sectionChecked);  
}
```

# Get the checked radio button

- Can declare an id="anId" in each radio button
- Use selection to see if each is "checked"

```
<input type="radio" id="radioOne" value="1">
```

```
. . .
```

```
<script>
```

```
function getTheData() {  
    element = document.getElementById("radioOne");  
    if (element.checked)  
        sectionChecked = element;
```

```
. . .
```

```
}
```

```
</script>
```

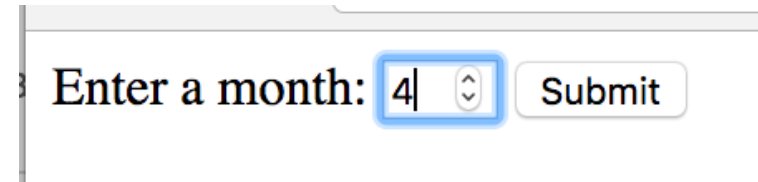
# One final new thing

- We can make it so all form elements must have values when the submit button is clicked with **required**

```
<form onsubmit="getData()">
  <input type="text" id="userID" required> Account Name<br>
  <input type="password" id="password" required> Password<br>
  <fieldset> <!-- fieldsets group things like a <div> -->
  <legend>Section</legend> <!-- legend also needed in lab -->
    1 <input type="radio" name="section" value="1"
      required> 11:00am<br>
    2 <input type="radio" name="section" value="2"
      required> 1:00pm<br>
    3 <input type="radio" name="section" value="3"
      required> 3:00pm
  </fieldset>
```

# In class Activity a.k.a Learning

*before*



Enter a month:

- Build a form so user can enter a month number in the range of 1..12 (use input type="number" min and max)
- Add a JS function to be called when the user clicks Submit
- Then set a season for the month number
  - 12, 1, 2: Winter
  - 3, 4, 5: Spring
  - 6, 7, 8: Summer
  - 9, 10, 11: Fall
- Make the input text field required
- Use Chrome and replace the entire html with this <h4>

*after*



**Month 4 is in Spring**

- If using Eclipse browser use `<form onsubmit="f(); return false;">`