



**COMP 307**  
Principles  
of Web  
Development

MCGILL UNIVERSITY

# COMP 307

## Principles of Web Development

### Lecture 7

#### Unit 2 – Frontend Internet Languages

#### Forms, Inspect & Graphics

[Contents](#)

Forms  
Canvas



**COMP 307**  
Principles  
of Web  
Development

# Class Outline

- HTML Forms
- Browser Inspect
- Graphics using HTML and JS

Contents

Forms  
Canvas



# Readings

- Internet and World Wide Web textbook
  - Chapters: 4.11
- Internet Resources
  - [https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)
  - [https://www.tutorialspoint.com/html/html\\_forms.htm](https://www.tutorialspoint.com/html/html_forms.htm)



**COMP 307**  
Principles  
of Web  
Development

# HTML Forms

Forms, Inspect & Graphics

## Contents

Forms  
Canvas



# Purpose

A simple and standard way to ask the user for information.

The information is converted into an **ASCII expression** and can then be passed as a query to a server.



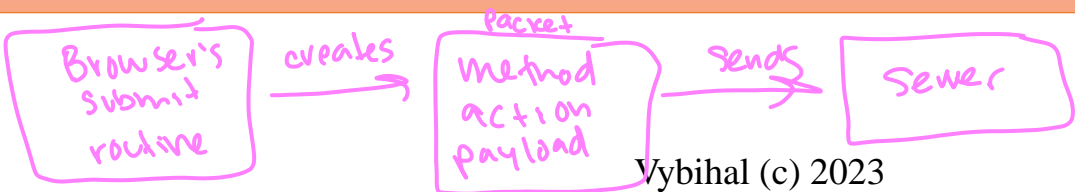
# Example

```
<form name="input" action="script.py" method="get" autocomplete="on">
  <b>Username:</b> <input type="text" name="user"> <br />
  Type of student: <br />
  <input type="radio" name="student" value="ugrad">Undergrad<br />
  <input type="radio" name="student" value="grad">Graduate<br />
  Graduating : <input type="checkbox" name="graduating"> <br />
  <input type="submit" value="Press Here">
</form>
```

*Handwritten notes:*

- what to do on submit* (pointing to `action="script.py"`)
- html5 defaults off* (pointing to `autocomplete="on"`)
- multiple choice → choose 1* (pointing to the radio buttons)
- calls script.py* (pointing to `type="submit"`)
- (three kind of)*
  - communication goes from browser to server
  - server returns an entire new webpage

After pressing submit, the ASCII expression: "user=bla&student=ugrad&graduating=true" *} data side of packet*





# Basic Form Tags

## Text input:

```
<input type="text" name="var2" size="25" maxlength="30">
```

```
<textarea name="feedback" rows=5 cols=20>
```

Initial text in box ← User deletes before adding their own

```
</textarea>
```

```
<input type="password" name="var3" size="25">
```

## Radio buttons: (only one active within named group) *single select*

```
<input type="radio" name="size" value="s"> short
```

```
<input type="radio" name="size" value="t"> Tall
```

## Checkboxes: (multiple can be active) *multiselect*

```
<input type="checkbox" name="bike"> Regular bike
```

```
<input type="checkbox" name="moto"> Motorcycle
```

## Buttons:

```
<input type="submit" value="Push Me">
```

```
<input type="reset" value="Clear Form"> deletes everything
```



# Basic Form Tags

## Dropdown:

```
<select name="pets" id="pet-select">  
  <option value="">--Please choose an option--</option>  
  <option value="dog">Dog</option>  
  <option value="cat">Cat</option>  
  <option value="hamster">Hamster</option>  
  <option value="parrot">Parrot</option>  
  <option value="spider">Spider</option>  
  <option value="goldfish">Goldfish</option>  
</select>
```

## Hidden parameters:

```
<input type="hidden" name="var1" value="stuff1">
```

*— will still be submitted to server*  
*user can't see it but it can be inspected*





# What is “hidden” good for?

COMP 307  
Principles  
of Web  
Development

It is useful when you want to inherit (or pass) data from one form to another. This will need support from your server code.

## Use cases:

- You are building a web-game. User walks through a dungeon. Each webpage is a different room. In one room you pick up an object. Then you want to walk to another room. You want the object you picked up to go with you.
  - Option 1 – write the object into a database (permanent- when you logout it is still there)
  - Option 2 – Write the object into a hidden tag (faster, easier, and cheaper for the server but not permanent)
- After the user logs in they get an ID number that says they are valid. You want the ID number to disappear after logout.

[Contents](#)

Forms  
Canvas



# HTML5 Form Tags

**COMP 307**  
Principles  
of Web  
Development

```
<input type="color" autofocus/>
```

```
<input type="date" />
```

```
<input type="datetime" />
```

```
<input type="datetime-local" />
```

```
<input type="email" placeholder="x@y.com">
```

```
<input type="month" />
```

```
<input type="number" min="0" max="10" step="1" value="4" />
```

```
<input type="range" min="0" max="20" value="10" />
```

```
<input type="search" placeholder="Query" />
```

```
<input type="tel" placeholder="(###) ###-####"  
pattern="\(\d{3}\) +\d{3}-\d{4}" required />
```

```
<input type="time" />
```

```
<input type="url" placeholder="http://bla.com" />
```

```
<input type="week" />
```

*ugly*

Contents

Forms  
Canvas



# Browser Inspect

- To help web developers, most browsers have a built-in “inspect feature”
- This also leads to proprietary issues for software developers (leading to the placing of code on the server instead of the browser)
- DEMO of inspect feature on Chrome...



**COMP 307**  
Principles  
of Web  
Development

graphics  
↳ JS  
or  
tag based

# Canvas

Forms, Inspect & Graphics

## Contents

Forms  
Canvas



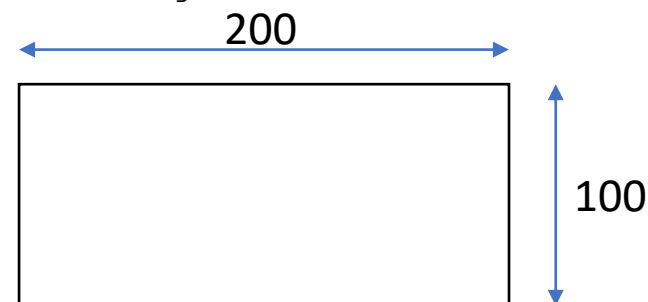
# What is the HTML Canvas?

It is a surface where JS can draw.

- A “container” for graphics.
- A “library” of methods for drawing shapes and pixel manipulation within the `<canvas>` area.

*uses JS*

Syntax: `<canvas id="myCanvas" width="200" height="100"></canvas>`





# Canvas and CSS

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100"
```

```
style="border:1px solid #000000;">
```

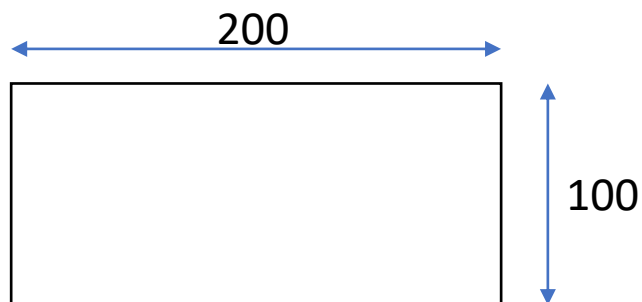
Your browser does not support the HTML canvas tag.

```
</canvas>
```

```
</body>
```

```
</html>
```

← alternative  
/canvas is  
on top of it



## Contents

Forms  
Canvas



# Canvas, CSS, and JS

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid  
#d3d3d3;">
```

Your browser does not support the HTML canvas tag.

```
</canvas>
```

```
<script>
```

```
var c = document.getElementById("myCanvas");
```

```
var ctx = c.getContext("2d");
```

```
ctx.moveTo(0,0); // lift "pencil" move to coordinate
```

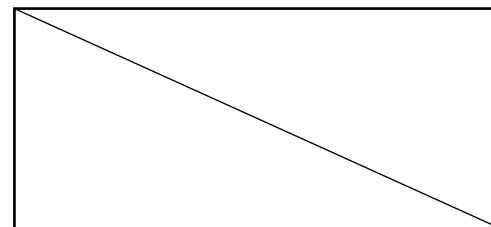
```
ctx.lineTo(200,100); // define destination coordinate
```

```
ctx.stroke(); // draw
```

```
</script>
```

```
</body>
```

```
</html>
```





# What does this do?

```
<script>
  var c = document.getElementById("myCanvas");
  var ctx = c.getContext("2d");

  ctx.beginPath();
  ctx.arc(95, 50, 40, 0, 2*Math.PI);
  ctx.stroke();
</script>
```

*begin*  
*end*  
*angle*

```
<script>
  var c = document.getElementById("myCanvas");
  var ctx = c.getContext("2d");

  ctx.font = "30px Arial";
  ctx.fillText("Hello World", 10, 50);
</script>
```

*start pos*





# What does this do?

```
<script>
  var c = document.getElementById("myCanvas");
  var ctx = c.getContext("2d");

  ctx.font = "30px Arial";
  ctx.strokeText("Hello World", 10, 50);
</script>
```

```
<script>
  var c = document.getElementById("myCanvas");
  var ctx = c.getContext("2d");
  var img = document.getElementById("dog");

  ctx.drawImage(img, 10, 10);
</script>
```

[https://www.w3schools.com/html/tryit.asp?filename=tryhtml5\\_canvas\\_tut\\_img](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_canvas_tut_img)



# Shading Examples

[https://www.w3schools.com/html/tryit.asp?filename=tryhtml5\\_canvas\\_tut\\_grad](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_canvas_tut_grad)

[https://www.w3schools.com/html/tryit.asp?filename=tryhtml5\\_canvas\\_tut\\_grad2](https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_canvas_tut_grad2)

## Clock

[https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas\\_clock\\_start](https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas_clock_start)

## Gaming Example

[https://www.w3schools.com/graphics/tryit.asp?filename=trygame\\_default\\_gravity](https://www.w3schools.com/graphics/tryit.asp?filename=trygame_default_gravity)

Using game controllers in Java Script: <https://beej.us/blog/data/javascript-gamepad/>

### Contents

Forms  
Canvas



Canvas -  
quality of image  
depends on your  
programming  
So ...

non-programming  
svg

# SVG

Forms, Inspect & Graphics

## Contents

Forms  
Canvas



# What is the HTML SVG?

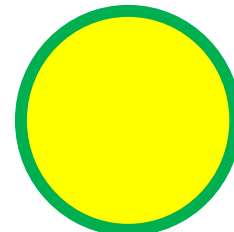
SVG is HTML scripting.

It is not programmable.

It uses a **vector-based** language to describe static graphics. (fast!) (scales well!) (limited ☹)

Syntax:

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"  
    fill="yellow" />  
</svg>
```



[https://www.w3schools.com/html/html5\\_svg.asp](https://www.w3schools.com/html/html5_svg.asp)



# Prepare for next class

- Assignments
  - Mini 3 due next class
- No labs this week
- On your own
  - Try to draw a stick figure person using both canvas and SVG