

HTML5 in practice

outline

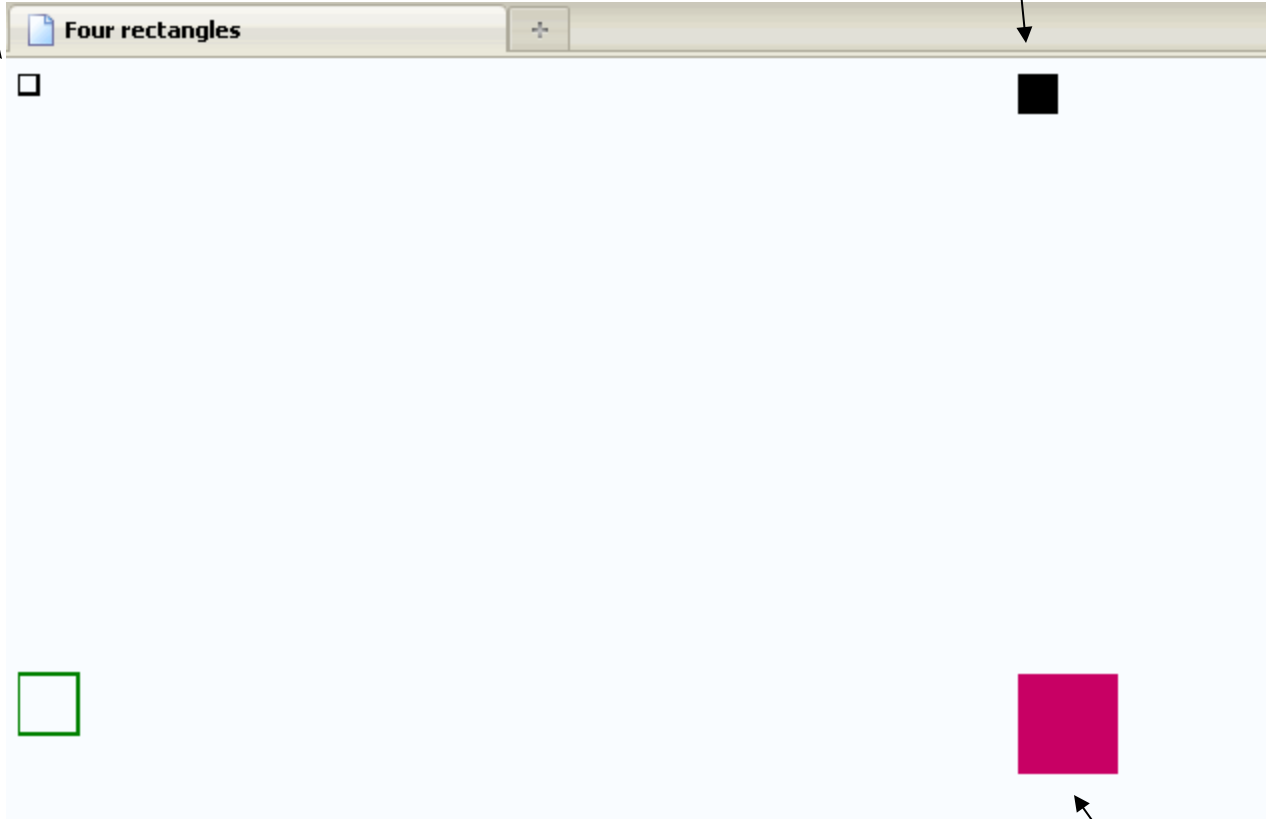
- Drawing on canvas
- Handling events
- Handling video
- Geolocation applications

drawing

- canvas element
- Use code to define a so-called context. Methods of this object do the work!
- Screen geometry: upper left corner is origin.
- Colors defined by red-green-blue values or a small set of named colors,
 - http://www.tutorialspoint.com/html5/html5_color_names.htm.
 - will show hexadecimal example later.
- stroke versus fill
- draw Rectangles
- <http://faculty.purchase.edu/jeanine.meyer/html5workshop/wkshopdrawing0.html>

0,0, default color, 10 by 10, stroke

500,0,default color,20 by 20, fill



0,300,green,30 by 30, stroke

rgb(200,0,100)

500,300, 50 by 50, fill

```
<!DOCTYPE html>
<html lang="en"><head><title>Four rectangles</title>
<meta charset="UTF-8"><script>
var ctx;
function init() {
    ctx =
document.getElementById('canvas').getContext('2d');
    ctx.lineWidth = 2;
    ctx.strokeRect(0,0,10,10);
    ctx.fillRect(500,0,20,20);
    ctx.strokeStyle = "green";
    ctx.fillStyle = "rgb(200,0,100)";
    ctx.strokeRect(0,300,30,30);
    ctx.fillRect(500,300,50,50); }
</script> </head>
<body onLoad="init();">
<canvas id="canvas" width="600" height="400"> Your
browser doesn't support the HTML5 element canvas.</
canvas> </body> </html>
```

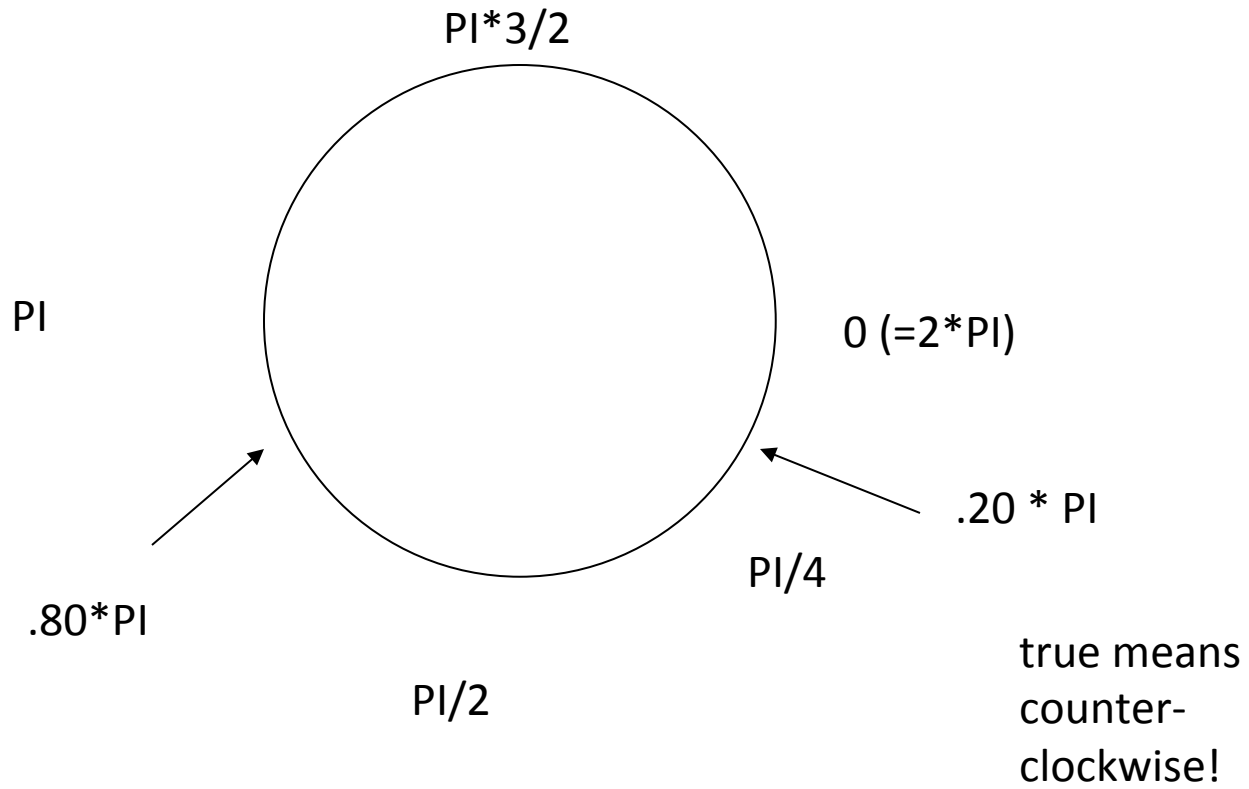
More comments

- Drawings are ...paint on the canvas.
- These rectangles are not objects to be moved or referenced later.
- Use `ctx.clearRect` method to erase.
- Need to do calculations to detect hits.
 - See memory game in book.
- Alternative is dynamic placement of html markup
 - See quiz, hangman.

Next drawing

- Paths created with arcs and line segments
- Arcs are portions of circles, created using radians in place of degrees. `Math.PI` is available for use. A complete circle can be drawn from 0 to $2 * \text{Math.PI}$ or $-\text{Math.PI}$ to `Math.PI`, etc.
- Arcs can be stroke or fill.
- <http://faculty.purchase.edu/jeanine.meyer/html5workshop/wkshopsmile.html>
- <http://faculty.purchase.edu/jeanine.meyer/html5workshop/wkshopfrown.html>

Angles



arcs

- `ctx.arc` (x of center, y of center, radius, starting angle, finishing angle, true for counter-clockwise)
- No drawing (ink) at the center! This is important when connecting arcs and lines.
- EXPERIMENT



4 distinct paths,
each made up
of 1 arc.

Default, "red"
and "brown"



Strategy

- Use variables with some variable values defined in terms of others.
- Circle face and two eyes. Smile is (partial) arc. Brown eyes and red smile.
- body element same as before.
 - You can add the code for this to your rectangles drawing.

```
var ctx;  
var headx = 100; //center of face x coord.  
var heady = 200; // center of face y coord.  
var headrad = 50; //radius of face  
var smileoffsetx=0; //smile center x is same as face  
var smileoffsety = 15; //smile center y further down  
var smilerad=20; // smile radius  
var eyeoffsety = -10; //eyes up from center  
var lefteyeoffsetx = -15; //left eye  
var righteyeoffsetx = -lefteyeoffsetx; //right  
var eyerad = 8; // eye radius
```

```
function init() {  
    ctx =  
    document.getElementById('canvas').getContext('2d');  
    ctx.lineWidth = 5;  
    ctx.beginPath();  
    ctx.arc(headx, heady, headrad,  
0, 2*Math.PI, true);  
    ctx.closePath();  
    ctx.stroke();  
    ...  
}
```

```
ctx.strokeStyle = "red";
ctx.beginPath();
ctx.arc(headx+smileoffsetx,heady+smileoffsety,
  smilerad,.80*Math.PI,.20*Math.PI,true);
ctx.stroke();
```

```
ctx.fillStyle = "brown";
  ctx.beginPath();
ctx.arc(headx+lefteyeoffsetx,heady
  +eyeoffsety,eyerad,
  0,2*Math.PI,true);
ctx.fill();
```

```
ctx.beginPath(); ctx.arc(headx
  +righteyeoffsetx,heady+eyeoffsety,eyerad,
  0,2*Math.PI,true);
  ctx.fill(); }
```

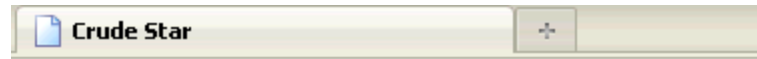
Comments

- The fill and stroke calls close the path.
- Also, can close a path with `closePath()`
- Using variables makes code more flexible and easier to see relationships.
- GO: draw arcs, changing colors, sizes, etc.
- NOTE: can draw non-circular ovals using transformations: `scale`. Check out the hangman game in book!

Next drawing: star

- For drawing lines (and arcs), think of moving a pencil versus drawing (preparing to draw) a line segment
 - nothing is drawn until the stroke or fill
- Use an array with coordinates for 5 points
- Use an array to hold names of 3 colors
- button element
- <http://faculty.purchase.edu/jeanine.meyer/html5workshop/wkshopdrawingstars.html>

opening screen



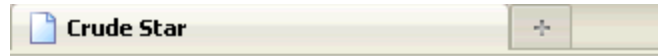
Make Star

after 1st press of button



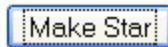
Make Star

after next press



Make Star

after next press



show body first

```
<body onLoad="init();">
```

```
<canvas id="canvas" width="600" height="400">
```

```
Your browser doesn't support the HTML5  
element canvas.
```

```
</canvas>
```

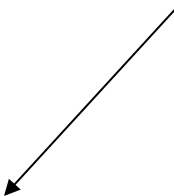
```
<button onClick="makestar();">Make Star </  
button>
```

```
</body>
```

```
</html>
```

variables (in script element)

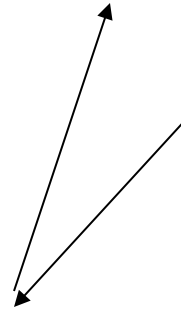
```
var ctx;  
var pts=[  
  //5 points for star: rough drawing  
  [100,35],  
  [60,10],  
  [20,35],  
  [35,100],  
  [85,100]  
];  
var colors=["red","white","blue"];  
  //used in succession  
var c=0;          // points to next color
```



variables (in script element)

```
var ctx;
```

```
var pts=[ //5 points for star: rough drawing  
  [100,35],  
  [60,10],  
  [20,35],  
  [35,100],  
  [85,100]  
];
```



```
var colors=["red","white","blue"];
```

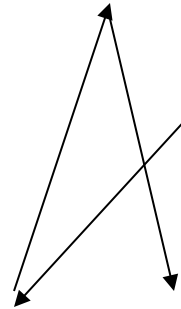
```
//used in succession
```

```
var c=0; // points to next color
```

variables (in script element)

```
var ctx;
```

```
var pts=[ //5 points for star: rough drawing  
  [100,35],  
  [60,10],  
  [20,35],  
  [35,100],  
  [85,100]  
];
```



```
var colors=["red","white","blue"];
```

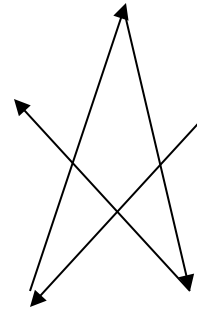
```
//used in succession
```

```
var c=0; // points to next color
```


variables (in script element)

```
var ctx;
```

```
var pts=[ //5 points for star: rough drawing  
  [100,35],  
  [60,10],  
  [20,35],  
  [35,100],  
  [85,100]  
];
```



```
var colors=["red","white","blue"];
```

```
//used in succession
```

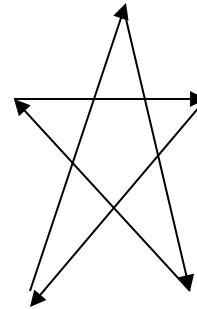
```
var c=0; // points to next color
```

variables (in script element)

```
var ctx;
```

```
var pts=[  
  drawing  
  [100,35],  
  [60,10],  
  [20,35],  
  [35,100],  
  [85,100]  
];
```

```
//5 points for star: rough
```



```
var colors=["red","white","blue"];  
  //used in succession  
var c=0;
```

```
// points to next color
```

```
function init() {
    ctx =
document.getElementById('canvas').getContext('2d'); }
function makestar() {
    ctx.clearRect(0,0,600,400);
    ctx.fillStyle=colors[c];
    c = c +1;
// can reduce to one line using colors[c++]
    c = (c<3)?c:0;
    ctx.beginPath();
    ctx.moveTo(pts[0][0],pts[0][1]);
    ctx.lineTo(pts[3][0],pts[3][1]);
    ctx.lineTo(pts[1][0],pts[1][1]);
    ctx.lineTo(pts[4][0],pts[4][1]);
    ctx.lineTo(pts[2][0],pts[2][1]);
    ctx.lineTo(pts[0][0],pts[0][1]);
    ctx.stroke();
//outline (necessary for white star!
    ctx.fill(); }
```

Comments

- Your assignment: do something with a button. It does not have to be a star.
- But do use `moveTo` and `lineTo`.
- You can combine with rectangles (separate from paths) and arcs (can combine with lines).
- Try `stroke` and `fill`
- Can include multiple `moveTo`
 - think of picking up your pen and moving to a new spot on the paper/canvas.

Fancier stars

- Code to draw star more precisely
- Position each star randomly on canvas.
 - **Add** star with each button press
- Increase number of colors.
 - http://www.tutorialspoint.com/html5/html5_color_names.htm
 - improve coding for robustness
- <http://faculty.purchase.edu/jeanine.meyer/html5workshop/wkshopprecisestars.html>

after many presses

Precise Stars, randomly placed



Make Star

Strategy

- reuse code for `makestar`, with modification
 - remove `clearRect` method
 - add a call to `buildstar` that re-creates the `pts` array
- add items to `colors` array
 - remove 3 from the code!
- position, size, and rotation of star created using calls to `Math.random`.

variables

```
var ctx;
```

```
var angle = 2*Math.PI/5;
```

```
var pts=[ ];
```

```
var
```

```
colors=["red", "white", "blue", "purple",  
"yellow", "teal"];
```

```
var c=0;
```



```
function makestar() {  
buildstar() ;  
ctx.fillStyle=colors[c++] ;  
c = (c<colors.length)?c:0;  
ctx.beginPath() ;  
ctx.moveTo(pts[0][0],pts[0][1]) ;  
ctx.lineTo(pts[3][0],pts[3][1]) ;  
ctx.lineTo(pts[1][0],pts[1][1]) ;  
ctx.lineTo(pts[4][0],pts[4][1]) ;  
ctx.lineTo(pts[2][0],pts[2][1]) ;  
ctx.lineTo(pts[0][0],pts[0][1]) ;  
ctx.stroke() ;  
ctx.fill() ;  
}
```

```
function buildstar() {  
  pts = [];  
  var x=500*Math.random(); //all these arbitrary  
  var y = 300*Math.random();  
  var r=50*Math.random();  
  var sangle = Math.random()*angle;  
  
  for(var i=0;i<5;i++) {  
    var a = i*angle + sangle;  
    var px = x+r*Math.cos(a);  
    var py = y-r*Math.sin(a);  
    pts.push([px,py]);  
  }  
  
}
```

Comments

- Lesson of these examples is that drawings are dynamic!
 - can be done under specific circumstances in a program
 - using different values
- Do this if there is time. Consider changing
 - names of colors
 - constants
 - draw something instead of or in addition to the star
 - ?

Next: coin flip

- Draw image from image file of head or tail on canvas where player clicks mouse
 - event handling: listen for mouse click
 - draw image made from external file
- Draw text (with directions) on canvas
- <http://faculty.purchase.edu/jeanine.meyer/html5workshop/wkshopcoinflip.html>

opening screen

Click on canvas to flip a coin.



after mouse click

Click on canvas to flip a coin.



Strategy

- Need to locate/identify file address for images
 - can be in same folder (use relative address) or use full URL
- Image objects with src attribute
- font set for context (ctx)
- event handling done for canvas element NOT context. Sets up call to function that has 1 parameter
- Does require browser specific code to get mouse coordinates. (Sigh...)
 - use technique of checking for presence of attribute

Strategy, cont.

- Fonts are from what is available on the client computer.
- draw outline (strokeRect) to show player where canvas is.
- Alternative to color names or rgb is hexadecimal.
 - use PhotoShop or Paint Shop Pro
- Note that my code makes adjustment to put middle of image where mouse was clicked.

variables

```
var ctx;  
var canvas1;  
var head = new Image();  
var tail = new Image();  
head.src = "head.gif";  
tail.src = "tail.gif";  
var coinwidth = 100;  
var coinheight = 100;
```

functions

```
function init() {  
    ctx =  
    document.getElementById('canvas').getContext  
    ('2d');  
    canvas1 =  
    document.getElementById('canvas');  
  
    canvas1.addEventListener('click', flip, false)  
    ;  
    ctx.font = "italic 20px Accent";  
    ctx.fillStyle = "#dd00ff";  
    ctx.strokeRect(0, 0, 600, 400);  
    ctx.fillText("Click on canvas to flip a  
    coin.", 10, 20);  
}
```

```
function flip(ev) {
var mx;
var my;
ctx.clearRect(0,0,600,400);
  if ( ev.layerX || ev.layerX == 0) { // Firefox
      mx= ev.layerX;
      my = ev.layerY;
  } else if (ev.offsetX || ev.offsetX == 0) { //
Opera
      mx = ev.offsetX;
      my = ev.offsetY;
  }
  if (Math.random()>.5) {
    ctx.drawImage(head,mx-50,my-50,coinwidth,coinheight); }
  else {
    ctx.drawImage(tail,mx-50,my-50,coinwidth,coinheight);}

ctx.strokeRect(0,0,600,400);
ctx.fillText("Click on canvas to flip a coin.",10,20); }
```

Comments

- You need to acquire two images to represent the coin faces.
 - download to same folder or use exact, complete address
- You can go into Word, or equivalent, to see available fonts.
 - more in book AND online.
- Go!

Background on audio & video

- This is **native** support (no need for plugins), no dependence on YouTube.
- Issue: Multiple formats (aka codecs) for each.
 - Different browsers recognize different formats.
 - Situation MAY improve: may be standardization.
 - Good news: the <audio> and <video> elements provide way to reference multiple files and **let** browser choose (though order does seem to be important).

Audio & video elements

```
<audio autobuffer>
```

```
<source src="crowdohh.ogg" />
```

```
<source src="crowdohh.mp3" />
```

```
</audio>
```

```
<video controls="controls">
```

```
<source src="sfire3.mp4" type='video/mp4;  
  codecs="avc1.42E01E, mp4a.40.2"'>
```

```
<source src="sfire3.theora.ogv" type='video/  
  ogg; codecs="theora, vorbis"'>
```

```
<source src="sfire3.webmvp8.webm"  
  type="video/webm; codec="vp8, vorbis"">
```

```
</video>
```

Our goal now

- Find and download video files.
- Use Miro Converter to produce other formats.
- Put audio into one of existing projects.
- Put video into one of existing projects.
- Build video bouncing ball.
- Look at book's **rock-paper-scissors** for playing specific sounds at specific times.
- Look at book's **quiz** for keeping video invisible and playing it only when quiz round complete.

Acquire video

- Make your own.
- Find on-line. Site <http://file2hd.com/> lets you specify a URL and choose from the videos to download.
 - Respect intellectual property!!!
 - Respect personal privacy (of friends & family, etc.)!!!

Produce all formats

- Produce formats for all browsers:
 - <http://www.mirovideoconverter.com/>
- Download the program.
- The original file can be in several different formats, including flv (Flash format, used by YouTube).
- Follow directions: should produce the 3 formats: mp4, theora.ogv, webmvp8.webm

Next: Video

- Put video element in your favorite sites or something else or ???
- Test using Firefox, Chrome and Safari.
- PATIENCE!!

Next: Bouncing video

- Two versions
 - move around video element
 - use `drawImage` to place video on canvas (works in FireFox)
 - Doing this makes it easier to draw mask. Otherwise, difficult to position.
- "Not quite ready for prime time". Need to put in code to wait until video all downloaded.

Strategy

- Use `setInterval` to invoke `[my] drawscene` function
 - re-position video.
 - Check against `[virtual]` walls. Apply angle of incidence equal angle of reflection (easy to do)
 - use `strokeRect` to draw box
- NOTE: the `loop` attribute did not work, so I set up event handling for video *ended* event.

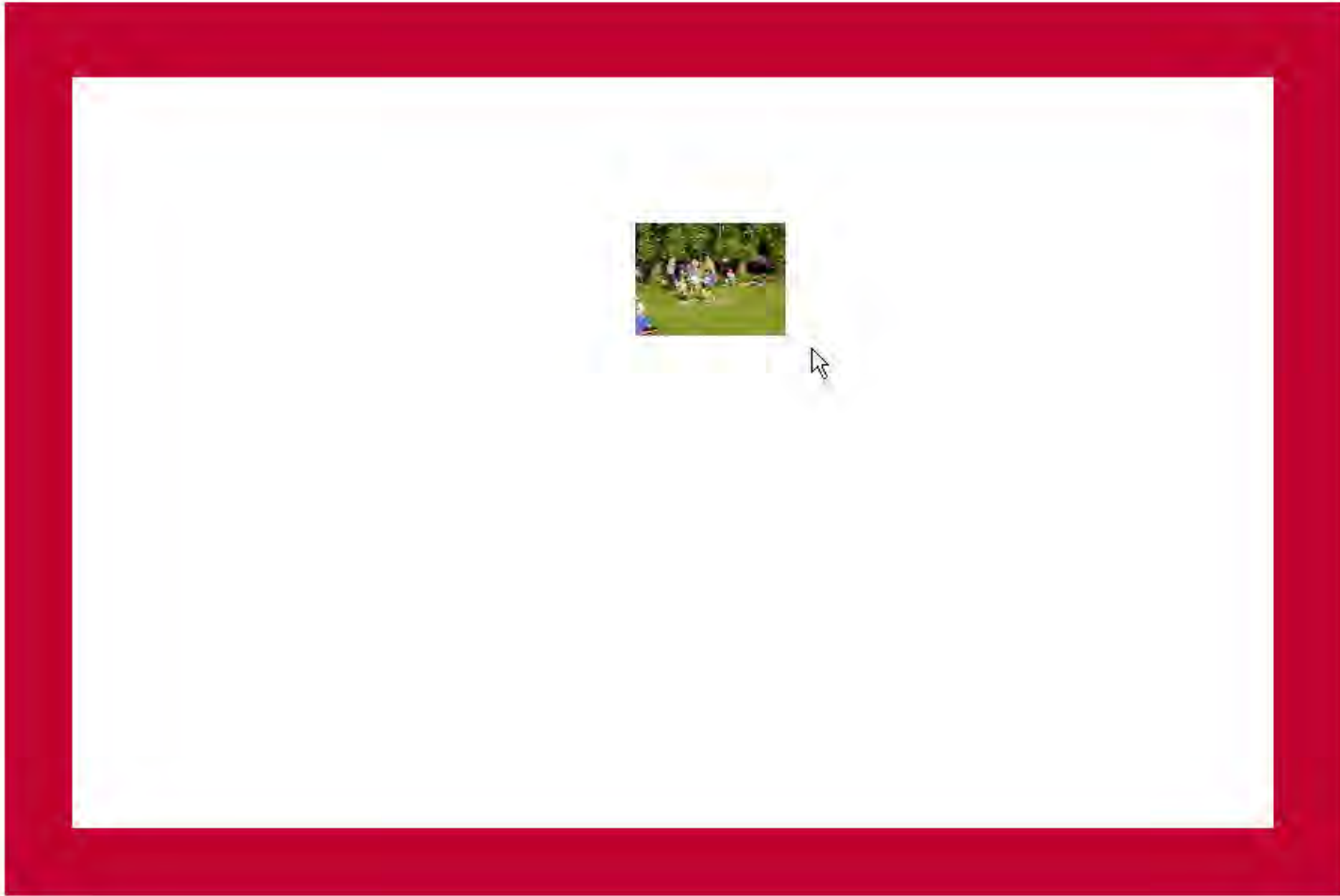
Function table

init	Called by action of onLoad in <body>
restart	... addEventListener in init
drawscene	... setInterval in init. This is different in two versions
moveandcheck	Called in drawscene

Bouncing video rectangle

- canvas and video distinct elements
- [http://faculty.purchase.edu/
jeanine.meyer/html5workshop/
videobounce1.html](http://faculty.purchase.edu/jeanine.meyer/html5workshop/videobounce1.html)

Video is in motion



Style section

Positioning and layering (my term).

Note: to change z-index by JavaScript, use `zIndex`.

```
<style>
```

```
#vid {position:absolute;  
  visibility:hidden; z-index: 0; }
```

```
#canvas {position:absolute; z-index:  
  10; }
```

```
</style>
```



```
<script type="text/javascript">
  var ctx;
  var cwidth = 900;    var cheight = 600;
  var ballrad = 50;
  var boxx = 20;      var boxy = 30;
  var boxwidth = 850;
  var boxheight = 550;
  var boxboundx = boxwidth+boxx-2*ballrad;
  var boxboundy = boxheight+boxy-2*ballrad;
  var inboxboundx = 0;
  var inboxboundy = 0;
  var ballx = 50;     var bally = 60;
  var ballvx = 2;     var ballvy = 4;
  var v;
```

```
function init() {
    ctx =
document.getElementById('canvas').getContext('2d
');
    v = document.getElementById("vid");
    v.addEventListener("ended", restart, false);
    v.style.left = ballx;
v.style.top = bally;
    v.width = 2*ballrad;
    v.height = 2*ballrad;
v.play();
    v.style.visibility = "visible";
    setInterval(drawscene, 50);
}
```

```
function restart() {  
    v.currentTime=0;  
    v.play();  
}
```

```
function drawscene() {
  ctx.lineWidth = ballrad;
  ctx.clearRect(boxx,boxy,boxwidth,boxheight);

  moveandcheck();

  v.style.left = ballx;
  v.style.top = bally;

  ctx.strokeStyle = "rgb(200,0,50)";

  ctx.strokeRect(boxx,boxy,boxwidth,boxheight);  /
  /box

}
```

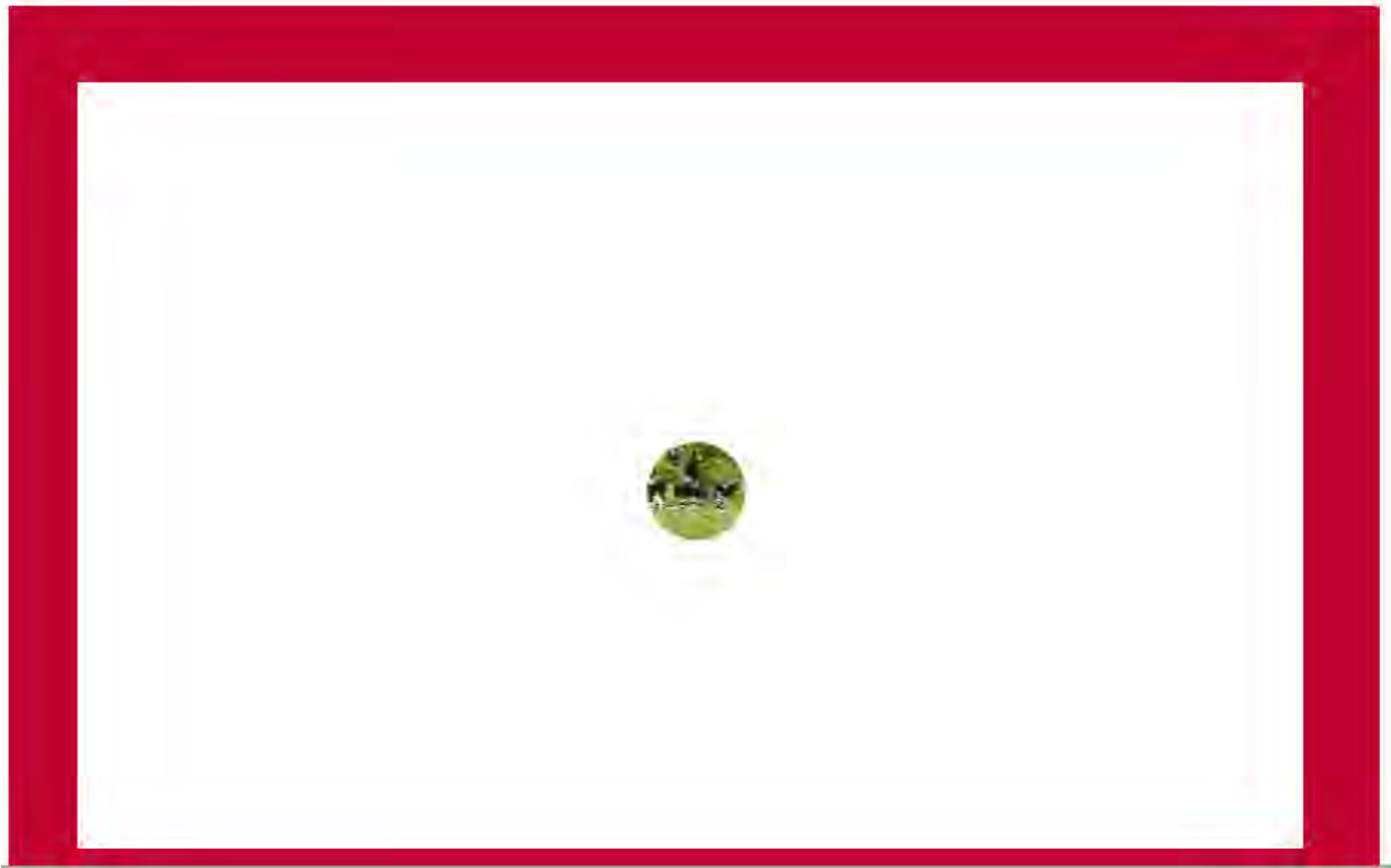
```
function moveandcheck() {
  var nballx = ballx + ballvx;
  var nbally = bally +ballvy;
  if (nballx > boxboundx) {
    ballvx =-ballvx;
    nballx = boxboundx; }
  if (nballx < inboxboundx) {
    nballx = inboxboundx
    ballvx = -ballvx; }
  if (nbally > boxboundy) {
    nbally = boxboundy;
    ballvy =-ballvy; }
  if (nbally < inboxboundy) {
    nbally = inboxboundy;
    ballvy = -ballvy; }
  ballx = nballx;
  bally = nbally; }
```

```
<body onLoad="init();">
<video id="vid" loop="loop" preload="auto">
<source src="joshuahomerun.mp4" type='video/
mp4; codecs="avc1.42E01E, mp4a.40.2"'>
<source src="joshuahomerun.theora.ogv"
type='video/ogg; codecs="theora, vorbis"'>
<source src="joshuahomerun.webmvp8.webm"
type='video/webm; codec="vp8, vorbis"'>
Your browser does not accept the video tag.
</video>
<canvas id="canvas" width="900" height="600">
This browser doesn't support the HTML5 canvas
element.
</canvas>
</body>
```

Bouncing video circle

- Works in Firefox.
- video used in drawImage.
- A mask is created: a box with a hole positioned over the video to produce a circular
- <http://faculty.purchase.edu/jeanine.meyer/html5workshop/videobounce3.html>

Video is in motion




```
<script>
  var ctx;
  var cwidth = 900;      var cheight = 600;
  var ballrad = 50;
  var boxx = 20;        var boxy = 30;
  var boxwidth = 850;
  var boxheight = 550;
  var boxboundx = boxwidth+boxx-2*ballrad;
  var boxboundy = boxheight+boxy-2*ballrad;
  var inboxboundx = 0;
  var inboxboundy = 0;
  var ballx = 50;       var bally = 60;
  var ballvx = 2;       var ballvy = 4;
  var maskrad;
  var v;
```

```
function init() {
    ctx =
    document.getElementById('canvas').getContext
    ('2d');
    v = document.getElementById("vid");

    v.addEventListener("ended", restart, false);
    // because loop doesn't work on FF
    v.width = v.videoWidth/3;
    v.height = v.videoHeight/3;
    videow = v.width;
    videoh = v.height;
    maskrad = .4*Math.min(videow, videoh);
    v.play();
    setInterval(drawscene, 50); }
```

```
function restart() {  
    v.currentTime=0;  
    v.play();  
}
```

```
function drawscene(){
    ctx.lineWidth = ballrad;
    ctx.clearRect(0,0,boxwidth+boxx,boxheight+boxy);
    ctx.fillStyle="rgb(255,255,255)"; //white
    moveandcheck();
    ctx.drawImage(v,ballx+boxx, bally+boxy, v.width,v.height);
    ctx.beginPath();
    ctx.moveTo(ballx+boxx,bally+boxy);
    ctx.lineTo(ballx+boxx+v.width,bally+boxy);
    ctx.lineTo(ballx+boxx+v.width,bally+boxy+v.height);
    ctx.lineTo(ballx+boxx,bally+boxy+v.height);
    ctx.lineTo(ballx+boxx,bally+boxy);
    ctx.arc(ballx+boxx+.5*v.width,bally+boxy+.5*v.height,maskrad,0,
        Math.PI*2,true);
    ctx.fill(); //draw white mask on top of video, letting just
        circle show
    ctx.strokeStyle ="rgb(200,0,50)";
    ctx.strokeRect(boxx,boxy,boxwidth,boxheight); //box
}
```

```
function moveandcheck() {
  var nballx = ballx + ballvx;
  var nbally = bally +ballvy;
  if (nballx > boxboundx) {
    ballvx =-ballvx;
    nballx = boxboundx; }
  if (nballx < inboxboundx) {
    nballx = inboxboundx
    ballvx = -ballvx; }
  if (nbally > boxboundy) {
    nbally = boxboundy;
    ballvy =-ballvy; }
  if (nbally < inboxboundy) {
    nbally = inboxboundy;
    ballvy = -ballvy; }
  ballx = nballx;
  bally = nbally; }
```

```
<body onLoad="init();">
<video id="vid" loop="loop" preload="auto">
<source src="joshuahomerun.mp4" type='video/mp4;
  codecs="avc1.42E01E, mp4a.40.2"'>
<source src="joshuahomerun.theora.ogv" type='video/
  ogg; codecs="theora, vorbis"'>
<source src="joshuahomerun.webmvp8.webm" type='video/
  webm; codec="vp8, vorbis"'>
```

Your browser does not accept the video tag.

```
</video>
```

```
<canvas id="canvas" width="900" height="600">
```

This browser doesn't support the HTML5 canvas element.

```
</canvas>
```

```
</body>
```

Next: Maps app

- Use Google Maps API to bring up map at current location.
- Respond to clicking by placing a marker and calculating distance.
- Provide way to change to fixed set of locations or the last marker.
- <http://faculty.purchase.edu/jeanine.meyer/html5workshop/geolocationdistance2.html>
 - need to give permission to Share Location
- Works in Chrome and Firefox. Does not work in Safari for Windows.

Opening screen



faculty.purchase.edu wants to track your physical location [Learn more](#)

Allow

Deny

Base location is your current geolocation



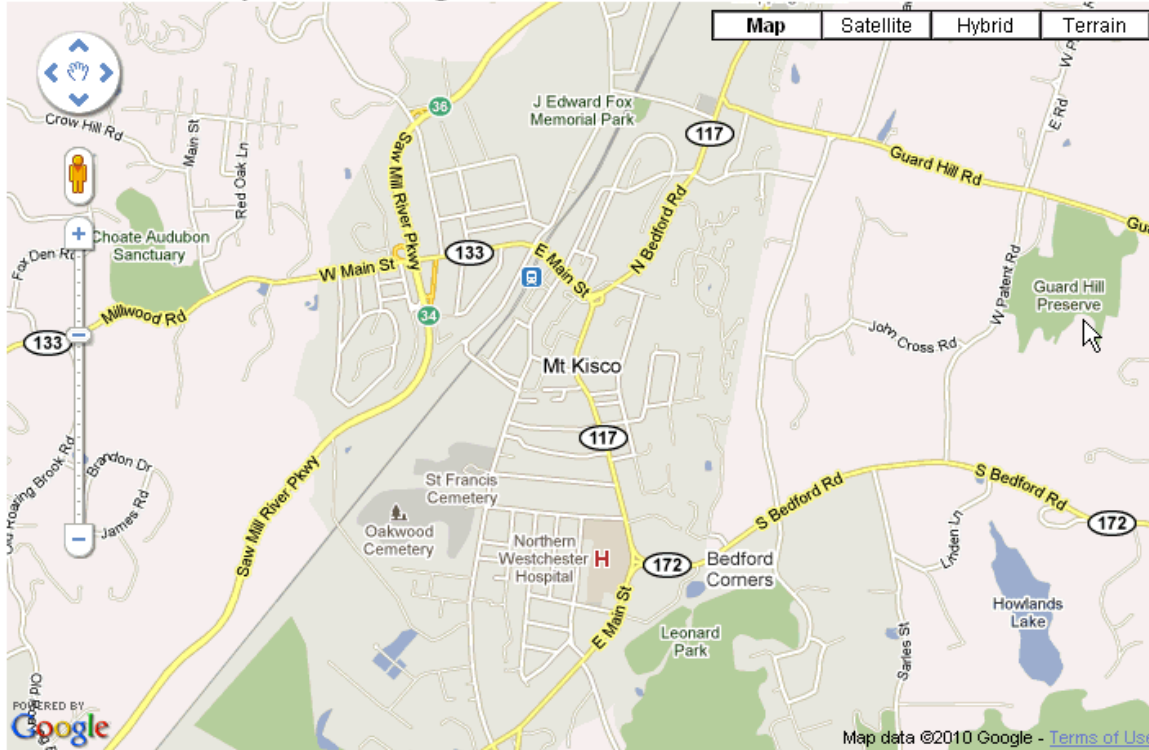
Change base location:

- CUNY
- Purchase College
- Illinois Institute of Technology

CHANGE

Brings up . . .

Base location is your current geolocation

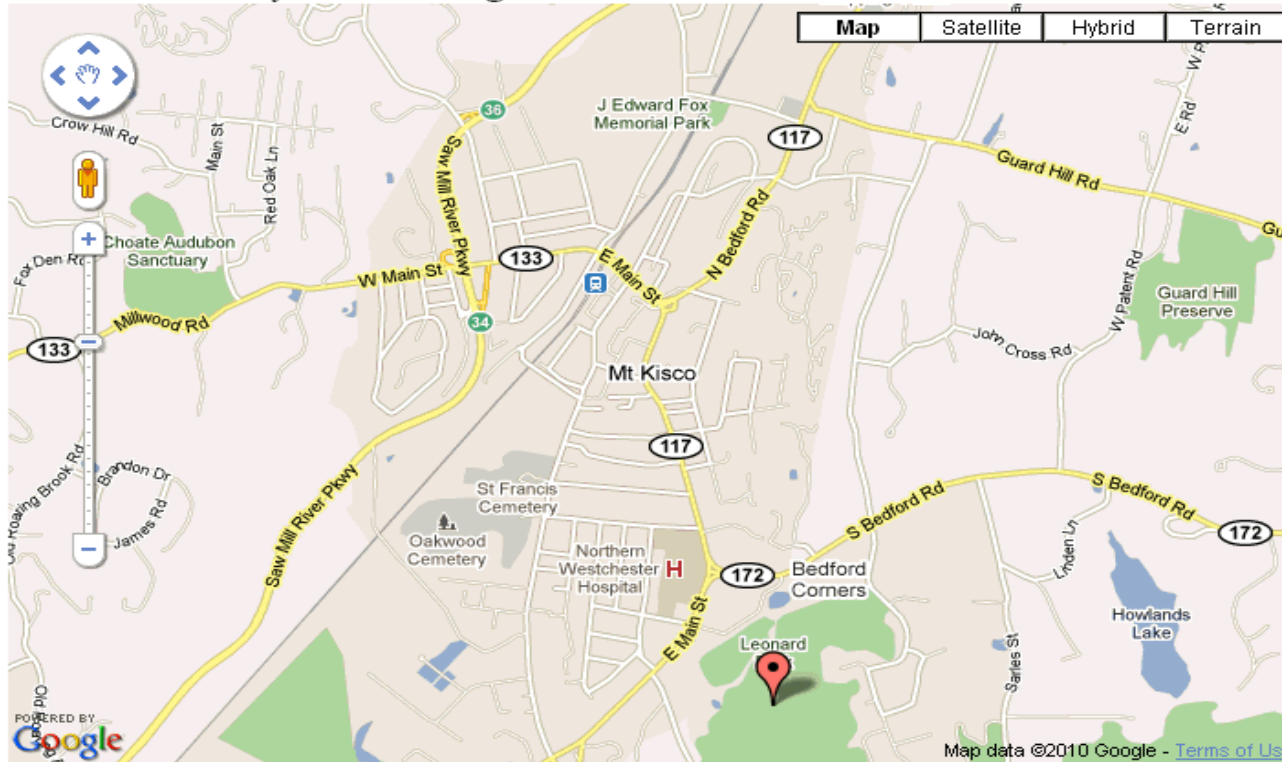


Change base location:

- CUNY
- Purchase College
- Illinois Institute of Technology

After click on map

Base location is your current geolocation



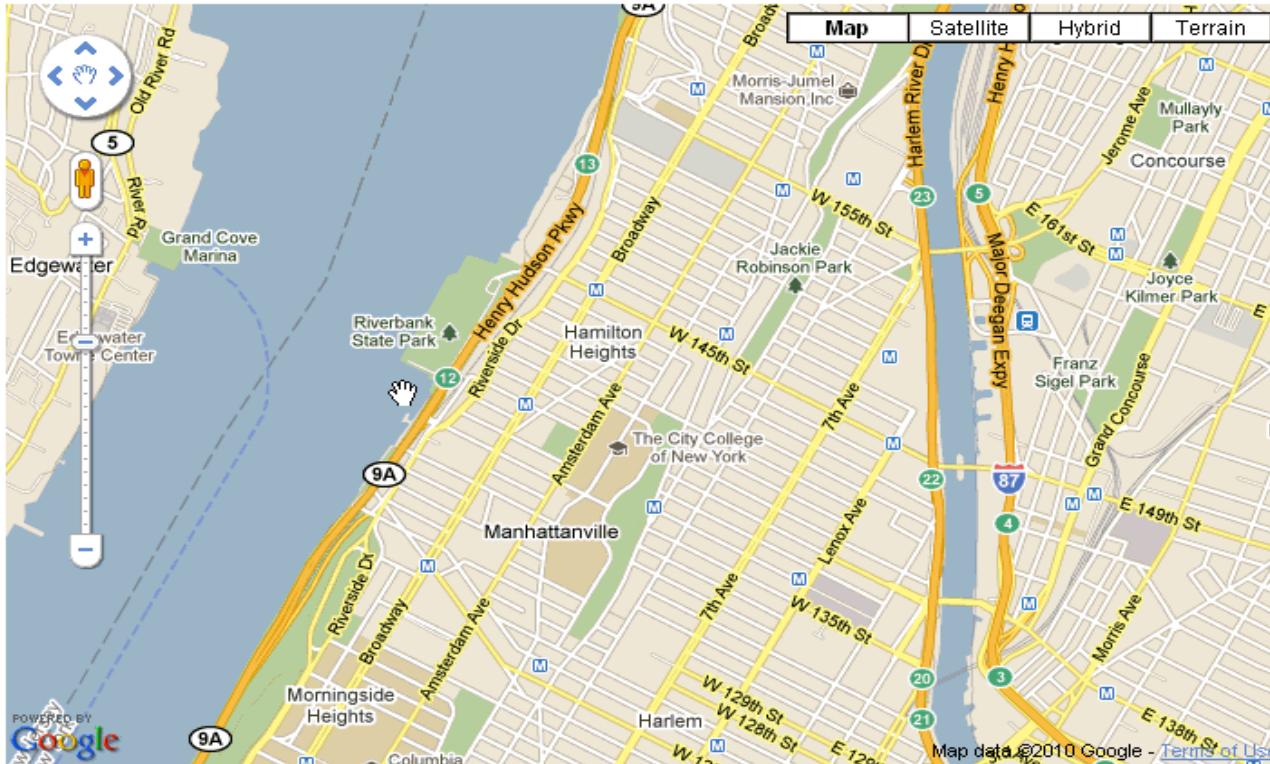
The distance from base to most recent marker is 0.93 miles.

Change base location:

- CUNY
- Purchase College
- Illinois Institute of Technology

After choose CUNY

Base location is CUNY



The distance from base to most recent marker is 0.93 miles.

Change base location:

- CUNY
- Purchase College
- Illinois Institute of Technology

CHANGE

Mapping

- Google Maps API (and other applications) defines positions using special latitude/longitude data object
- Access to Google Map is created for a place in the HTML document, using specified map options
- HTML has a specification for doing geolocation.
 - `navigator.geolocation` produces latitude and longitude values

How to get positions?

- Google Maps

- get to a map

- Browser location

- ```
javascript:void(prompt('',gApplication.getMap().getCenter()));
```

OR

- Click on green beaker and enable the drop latlng marker

- right click then normal click

# My program

- Tries to use the geolocation
- Gives user option to pick base location
- User can click on map and find distance from base center.
- Can change base to last clicked on position.
- General access to Google Maps features.

# Basics

- `if (navigator.geolocation)` checks if this object exists. Does NOT cause any errors.

```
if (navigator.geolocation) {
 navigator.geolocation.getCurrentPosition
 (
 handler, problemhandler
);
}
```

- `handler` when things go okay.  
`problemhandler` when there are errors, including user deciding not to share location.

# Create/Access Google Maps

- `map = new google.maps.Map(document.getElementById("place"), myOptions);` brings up Google Maps in the div with id "place" using the parameters in myOptions.

```
<div id="place" style="width:600px; height:400px"></div>
```

- NOTE: use of % for width and height did not work when `<!DOCTYPE html>` used.



# style, external script

```
<style>
```

```
header {font-family:Georgia,"Times New
Roman",serif;
font-size:20px;
display:block;
}
```

```
</style>
```

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/
js?sensor=false"></script>
```

```
<script language="Javascript">
```

# init() code

```
function init() {
 if (navigator.geolocation) {
 navigator.geolocation.getCurrentPosition
 (handler,problemandler);
 }
 else {
 if (document.getElementById("place")) {
 document.getElementById("place").innerHTML
 = "I'm sorry but geolocation services are
 not supported by your browser";

 document.getElementById("place").style.color
 = "#FF0000";
 } } }
}
```

```
function handler(position) {
 var mylat = position.coords.latitude;
 var mylong =
 position.coords.longitude;
 makemap(mylat, mylong);
}
```

# error handling

```
function problemhandler(prob) {
 switch(prob.code) {
 case 1:
 document.getElementById("place").innerHTML =
 "User declined to share the location
 information.";
 break;
 case 2:
 document.getElementById("place").innerHTML =
 "Errors in getting base location.";
 break;
 case 3:
 document.getElementById("place").innerHTML =
 "Timeout in getting base location."; }
 document.getElementById("header").innerHTML
 = "Base location needs to be set!"; }
```

# variables

```
var listener;
var map;
var markersArray = [];
var blatlng;
var myOptions;
var locations = [
 [35.1494444, -90.0488889, "Memphis,
 TN"], [41.04796, -73.70539, "Purchase
 College/SUNY"],
 [41.878928, -87.641926, "IIT"]
];
```

# create/access map

```
function makemap(mylat, mylong) {
 blatlng = new
 google.maps.LatLng(mylat,mylong);
myOptions = {
 zoom: 14,
 center: blatlng,
 mapTypeId: google.maps.MapTypeId.ROADMAP};
 map = new
 google.maps.Map(document.getElementById("place"), myOptions);
 listener =
 google.maps.event.addListener(map, 'click',
 function(event) {
 checkit(event.latLng);});
}
```

# response to click on map

```
function checkit(clatlng) {
 var distance = dist(clatlng,blatlng);
 distance = Math.floor((distance+.005)*100)/
 100;
 var distanceString = String(distance)+"
 miles";
 marker = new google.maps.Marker({
 position: clatlng,
 title: distanceString,
 map: map });
 markersArray.push(marker);
 document.getElementById("answer").innerHTML
 =
 "The distance from base to most recent
 marker is "+String(distance) +" miles."; }
```

# distance function

```
function dist(point1, point2) {
 //spherical law of cosines
 //var R = 6371; // km
 var R = 3959; // miles
 var lat1 = point1.lat()*Math.PI/180;
 var lat2 = point2.lat()*Math.PI/180 ;
 var lon1 = point1.lng()*Math.PI/180;
 var lon2 = point2.lng()*Math.PI/180;
 var d =
 Math.acos(Math.sin(lat1)*Math.sin(lat2) +
 Math.cos(lat1)*Math.cos(lat2) *
 Math.cos(lon2-lon1)) * R;
 return d;
}
```



# change base using radio buttons

```
function changebase() {
 for(var i=0;i<locations.length;i++) {
 if (document.f.loc[i].checked) {
 makemap(locations[i]
[0],locations[i][1]);

document.getElementById("header").innerHTML
= "Base location is "+locations[i][2];
 }
 }
 return false;
}
```

```
</script> </head> <body onload="init();">
<header id="header">Base location is your current
 geolocation</header>
<div id="place" style="width:600px; height:
 400px"></div>
<div id="answer"></div>
Change base location:

<form name="f" onSubmit=" return changebase();">
 <input type="radio" name="loc" /> Memphis

 <input type="radio" name="loc" /> Purchase
 College

 <input type="radio" name="loc" /> Illinois
 Institute of Technology

 <input type="submit" value="CHANGE">
</form> </body> </html>
```

# Variation

- Geolocation returns accuracy and, maybe, other information, including altitude.
- These applications mark the center with a red x and display other information
  - <http://faculty.purchase.edu/jeanine.meyer/html5workshop/geolocation.html>
  - <http://faculty.purchase.edu/jeanine.meyer/html5workshop/geolocationkm.html>
    - Note: accuracy is given in meters in both cases

# Critical code

- Uses custom image for marker

```
var xmarker = "x1.png";
```

```
...
```

```
marker = new google.maps.Marker({
 position: blatlng,
 title: "center",
 icon: xmarker,
 map: map });
```

# getCurrentPosition

- 3<sup>rd</sup> parameter to getCurrentPosition call

```
positionopts = {enableHighAccuracy: true} ;
```

...

```
navigator.geolocation.getCurrentPosition(handler,
 problemhandler, positionopts);
```

- Add form coutput for output

```
document.coutput.lat.value = mylat;
```

```
document.coutput.lon.value = mylong;
```

```
document.coutput.acc.value =
position.coords.accuracy;
```

```
document.coutput.alt.value =
position.coords.altitude;
```

```
document.coutput.altacc.value =
position.coords.altitudeAccuracy;
```

```
document.coutput.heading.value =
position.coords.heading;
```

```
document.coutput.speed.value = position.coords.speed;
```

# Next application: persistent storage

- Normal situation: no changes to client computer beyond downloaded files.
- **cookies** invented (with a pleasing name) to be files associated with each browser to be used only by same server site.
  - convenience: IDs and passwords, preferences, etc.
  - behavioral marketing!
- Early HTML supported cookies. localStorage is a new variation.
- CAUTION: Firefox requires the program to run on a server! Chrome allows running locally.
- <http://faculty.purchase.edu/jeanine.meyer/html5workshop/geolocationdistance4.html>

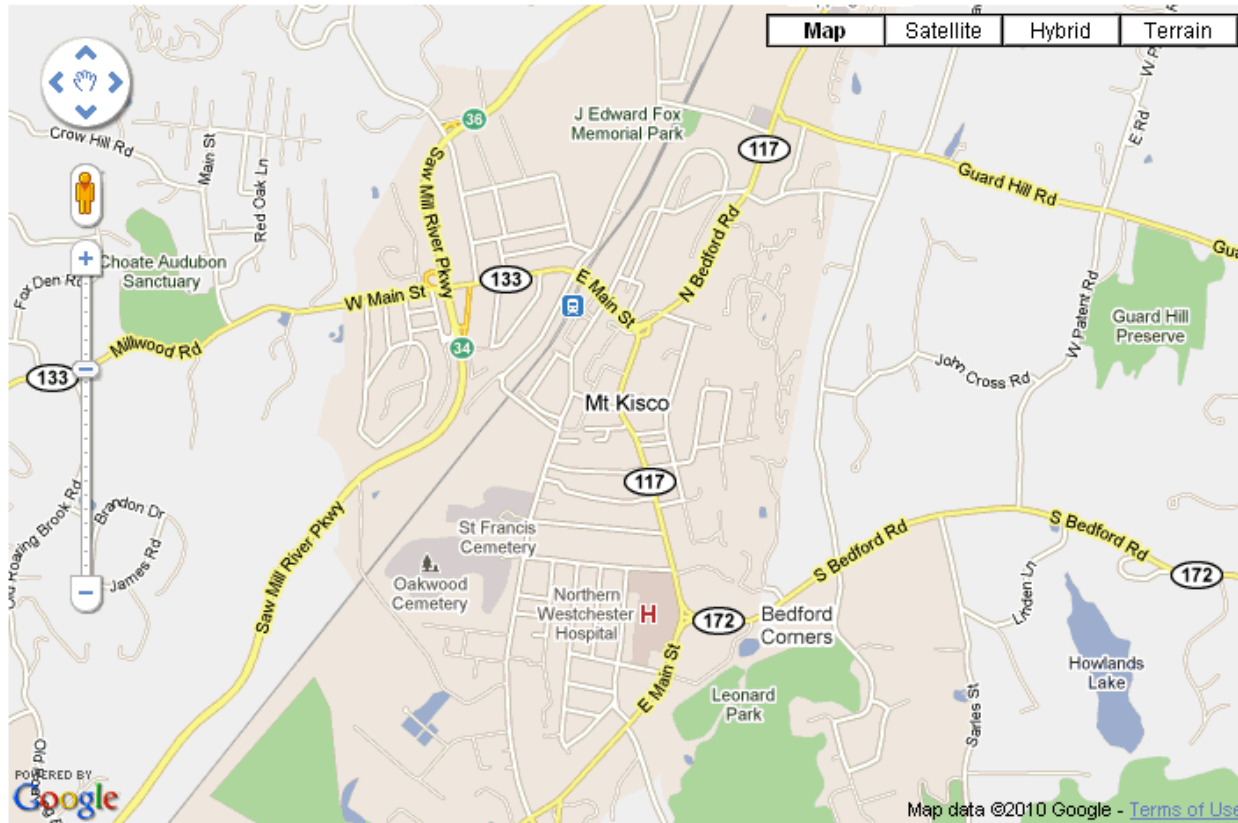
# Opening (after permission)

Base location is your current geolocation

Store base.

Restore last base.

Change base location to last marker.

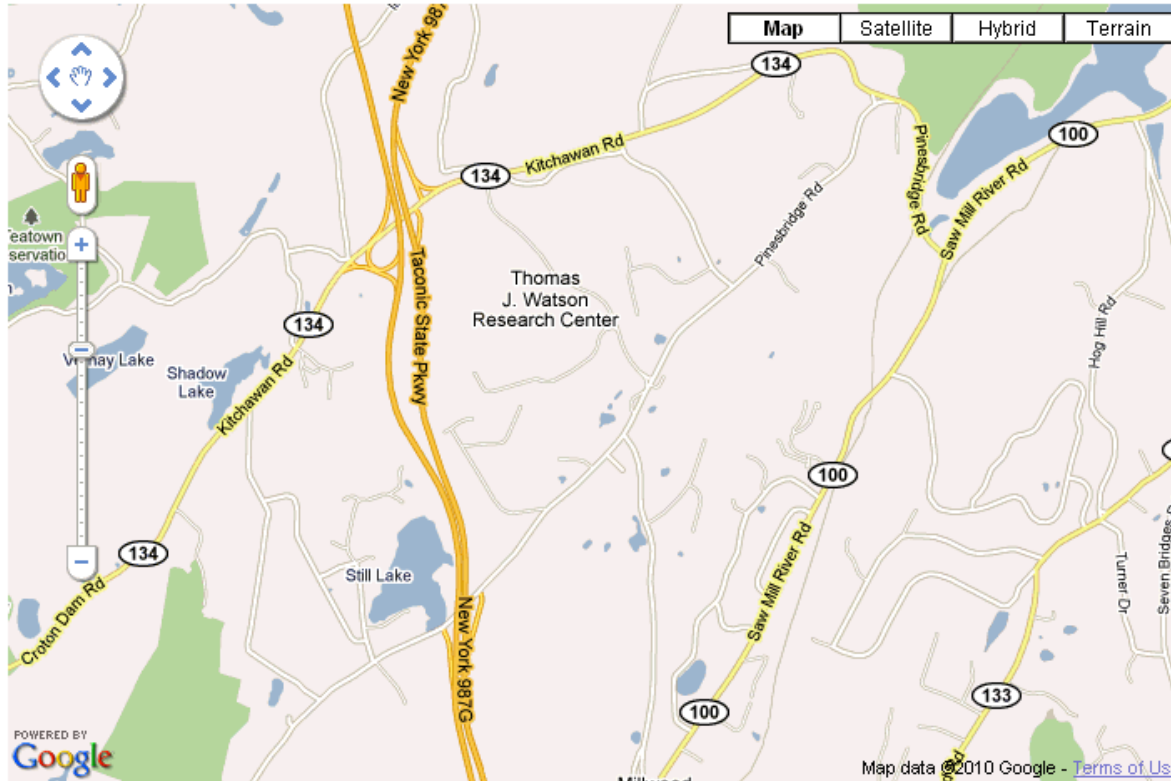


- CUNY
- Purchase College
- Illinois Institute of Technology

CHANGE

# New base

Base location is your current geolocation



The distance from base to most recent marker is 3.7 miles.

- CUNY
- Purchase College
- Illinois Institute of Technology



# Objective: add to maps app

- 3 buttons: store base, retrieve base stored, change base to last marker
- localStorage used name-value pairs.
- Do error checking!
  - check for ability to do localStorage
  - check if there is a stored time.

# Strategy

- Three buttons, invoking `store()`, `retrieve()`, and `changebasetomarker()`
- Use `try { } catch(e) { }`. The code in `try` will NOT trigger an error, but if there is one, will go to `catch`.
- Also use `typeof(localStorage)` to test if this is defined.

```
<button onClick="javascript:store();">Store
base. </button>

<button
onClick="javascript:retrieve();">Restore
last base. </button>

<button
onClick="javascript:changebasetomarker();">C
hange base location to last marker. </
button>

```

```
function store() {
 if (typeof(localStorage) == "undefined") {
 alert("Browser does not recognize HTML local
storage.");
 }
else { try {
 localStorage.setItem("baselat",blatlng.lat());
 localStorage.setItem("baselng",blatlng.lng());
 localStorage.setItem("basename",basename);
 }
catch(e) {
 alert("Error with use of local storage: "+e);}
}
return false; }
```

```
function retrieve() {
 if (typeof(localStorage) == "undefined") {
alert("Browser does not recognize HTML local storage.");
 }
else { try {
 oldlat= localStorage.getItem("baselat");
 oldlng = localStorage.getItem("baselng");
 oldname = localStorage.getItem("basename");
 if (oldlat==null) {
 alert("No base stored");}
 else {makemap(Number(oldlat),Number(oldlng));
 basename = oldname;
 document.getElementById("header").innerHTML = "Base
location is "+basename; } }
catch(e) {
 alert("Error with use of local storage: "+e);} }
return false; }
```

# changes base to marker

```
function changebasetomarker() {
 if (lastmarker!="undefined") {

 makemap(lastmarker.lat(),lastmarker.ln
g());
 basename = "marker";
 }
}
```

# Comments

- Error checking good!
- Many GIS programs with common/similar features
- Need to determine where information goes
  - my locations array kept information in my JavaScript