

# HELP!

I just wanna program something, not go on an acronym tour!



DHTML XHTML HTML

AJAX CSS HTTP

JavaScript C++ Java

.NET C++ Croquet

Smalltalk C# Squeak

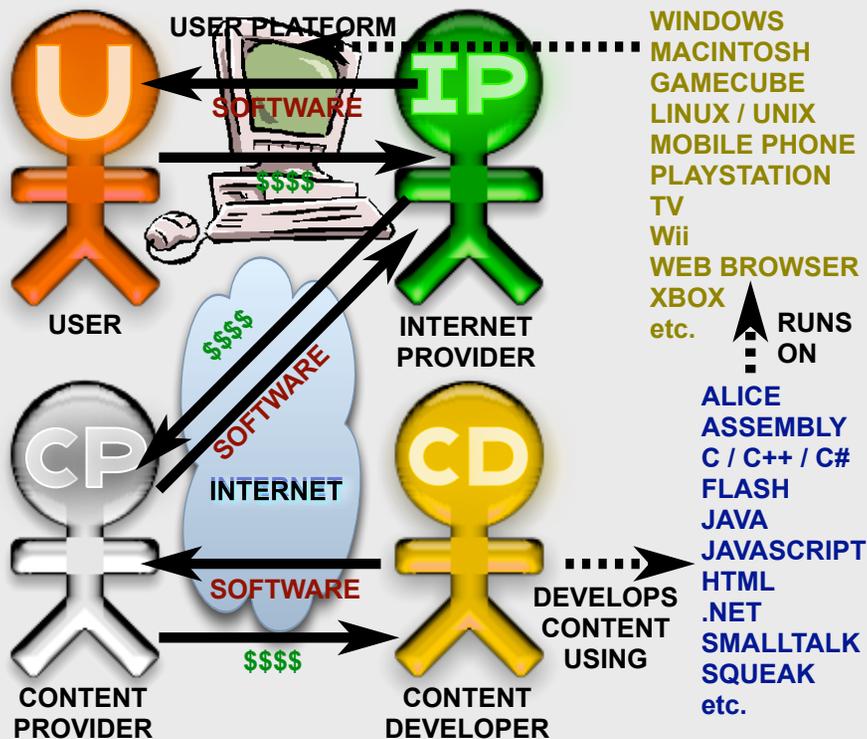
Flash SVG Perl

# How do I...

Now that's more like it!



## Create software?



- Edit
- source code
- Compile
- binary
- Run



# EXAMPLE

## How do I...

I thought Java was coffee...

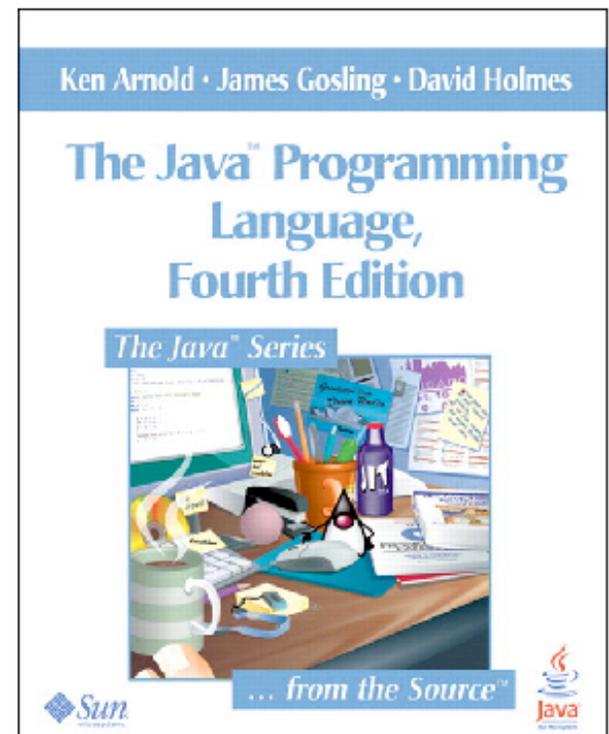
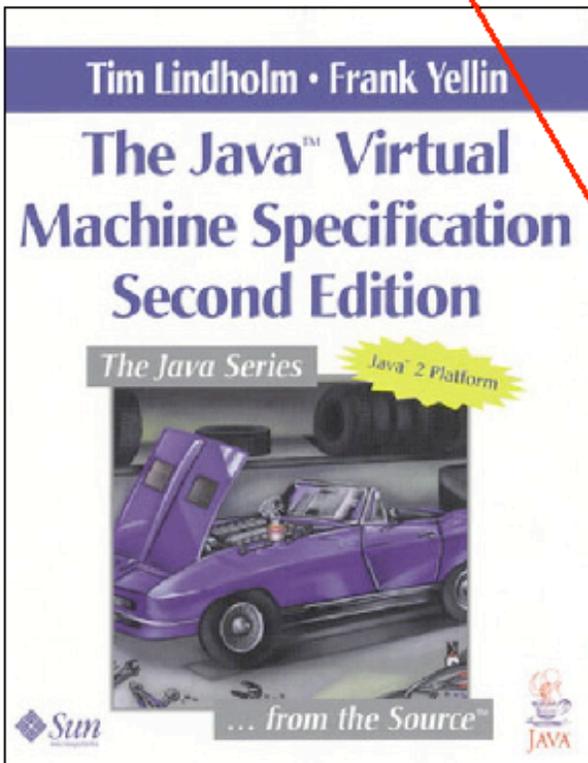
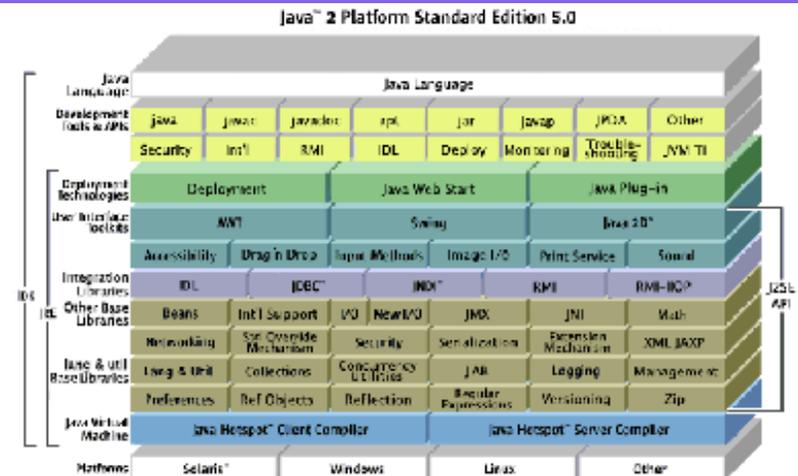
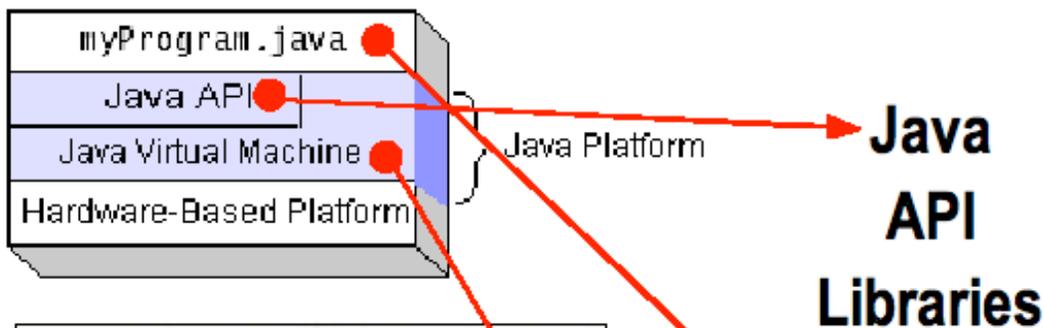


## Create a Java Program?

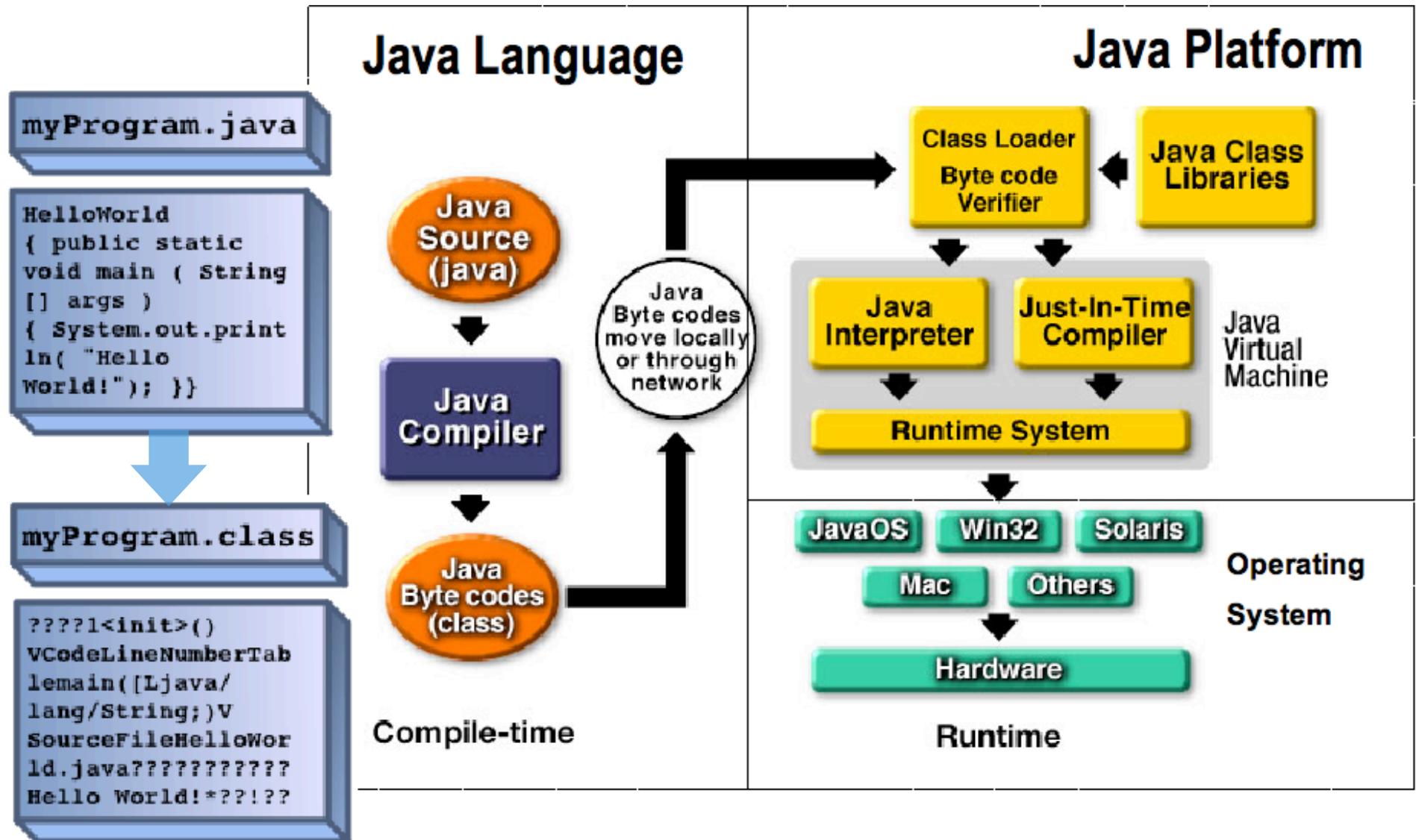
- Create program source code
  - `edit myProgram.java`
- Compile source code to binary
  - `javac myProgram.java`
- Run binary program
  - `java myProgram`



# EXAMPLE: Java Platform



# EXAMPLE: Java Program



# EXAMPLE: Java Program

<http://www.tiltitation.com>

**64 Levels!**  
**6 Unlockable balls!**  
**Only \$6.99**

## FEATURES

- 64 Fun Packed Levels
- Life-like Physics
- Realtime 3D Graphics
- 6 Unlockable Balls
- Online and Local Best Times
- Packed with Features and Toys
- Joypad Support
- Fun Sound Effects and Easy Listening Music
- Non-violent and Fun for All Ages

## TILTITATION?

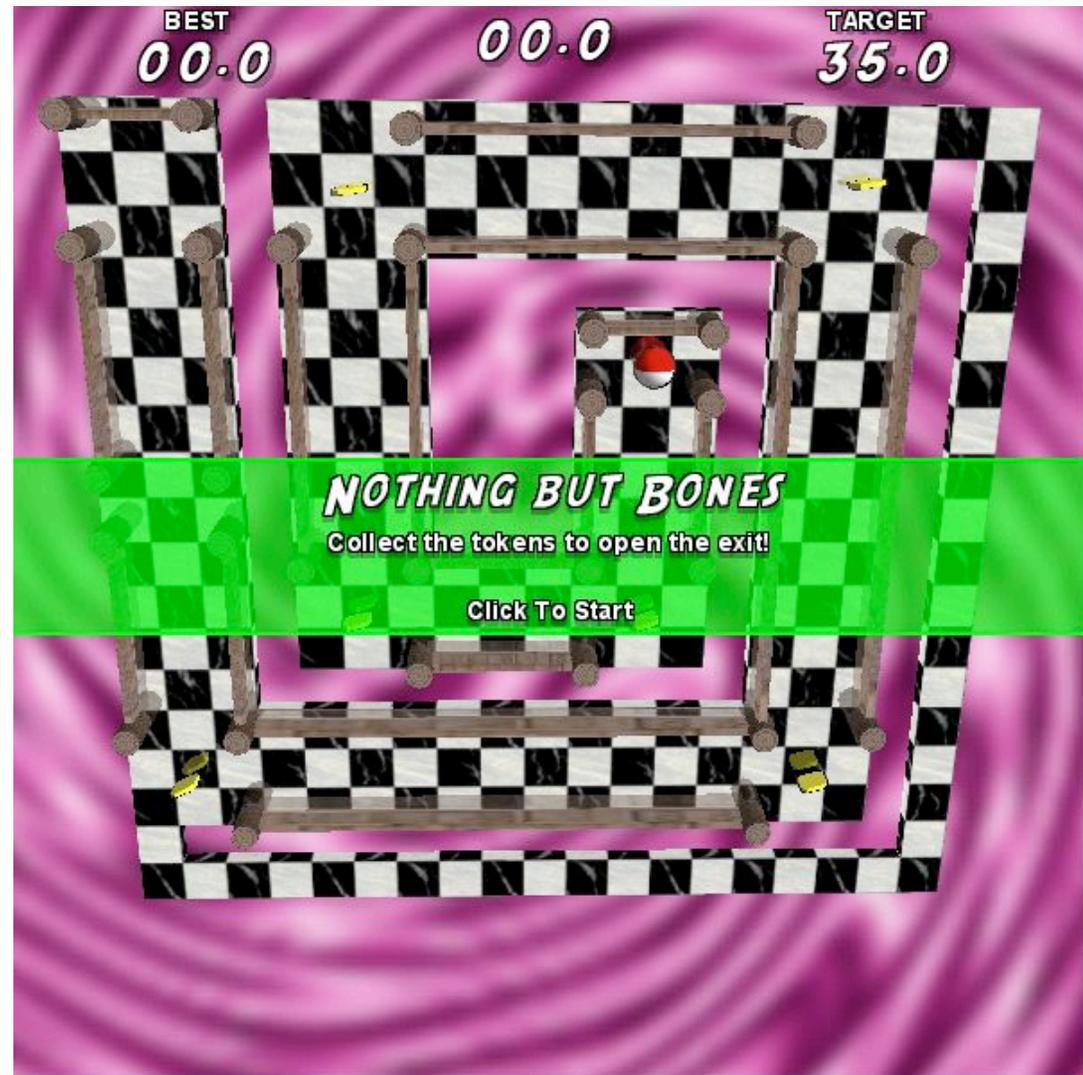
Tilt the board to guide your ball through the maze to the exit collecting the gold tokens along the way. Sound easy? Why not play it and tell us that after! Bounce, fall and dodge your way through over 50 levels of challenging fun.

 Download for Windows (6MB)

 Download for MacOSX (4MB)

 Download for Linux (9MB)

 Play via Webstart (3MB)

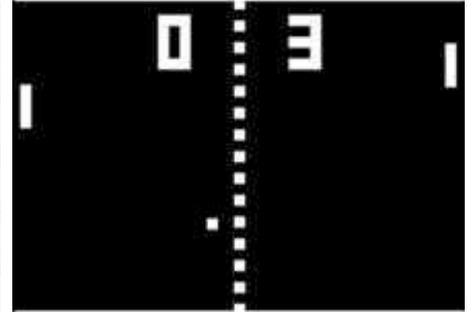
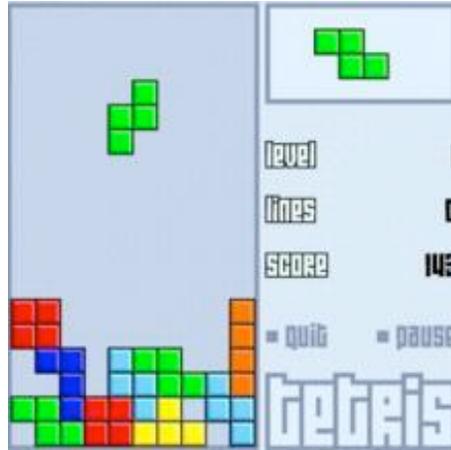


Subscribe for Tiltitation News and Updates!

Email

Subscribe

# EXAMPLE: Flash Program



[http://www.](http://www.80smusiclyrics.com/games.shtml)

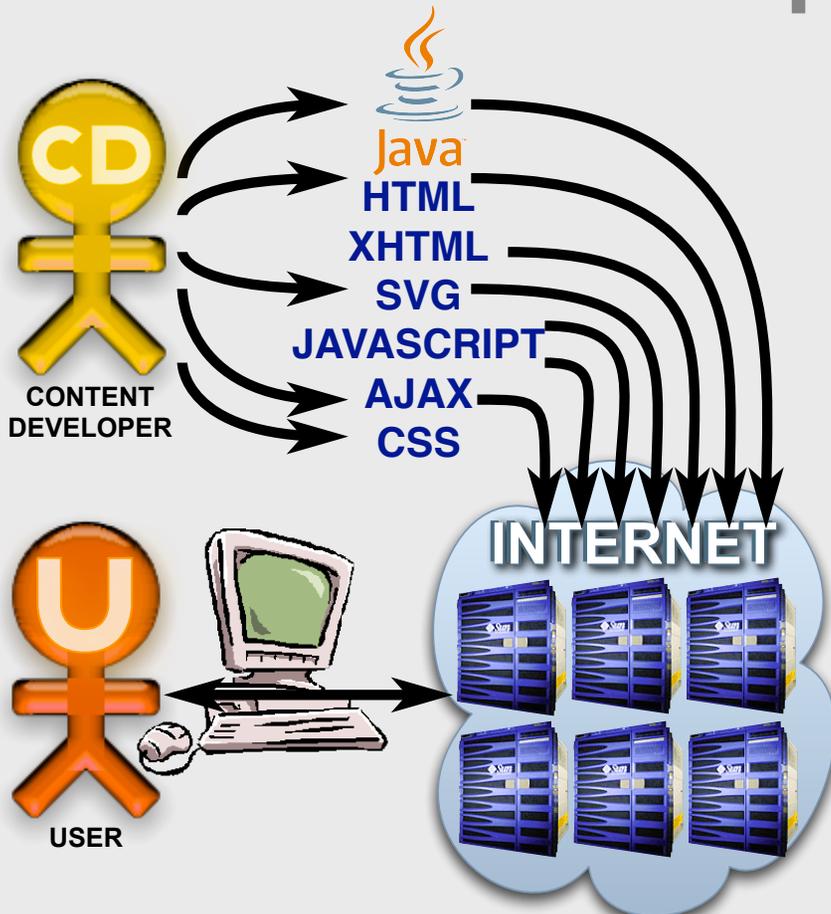
[80smusiclyrics.com/games.shtml](http://www.80smusiclyrics.com/games.shtml)

# How do I...

What a tangled web we weave



## Create a web page?



- Edit
- page source
- Browse
- web page



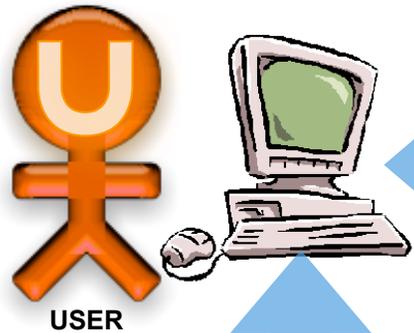
CONTENT DEVELOPER

WEB CONTENT



USER

# EXAMPLE: Web Platform



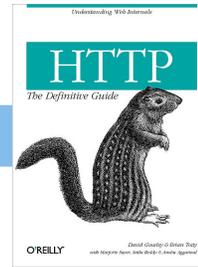
USER

HTML

HTTP

BROWSER

Internet Explorer,  
Mozilla, Firefox,  
Safari,  
etc.

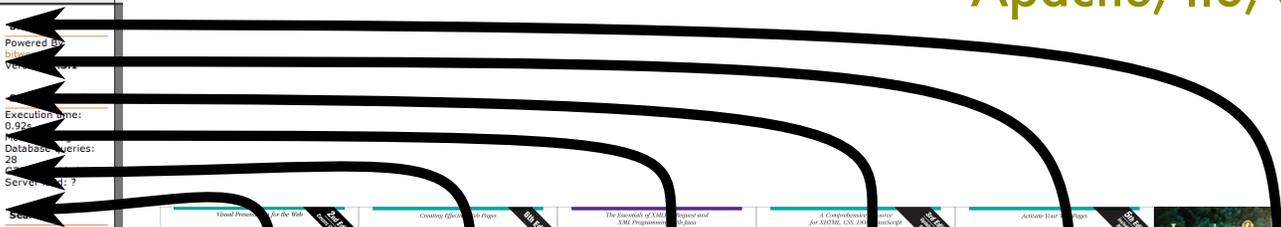
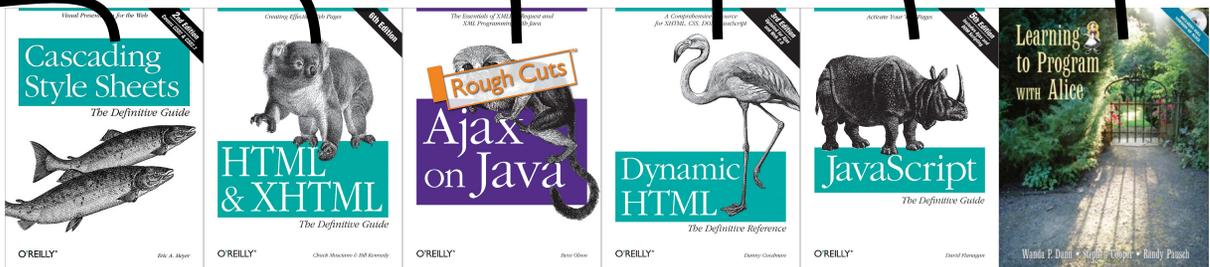
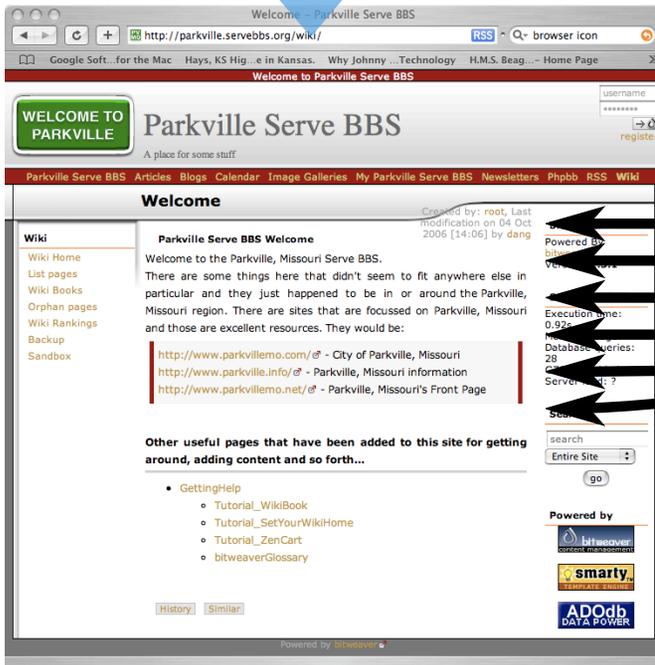


INTERNET

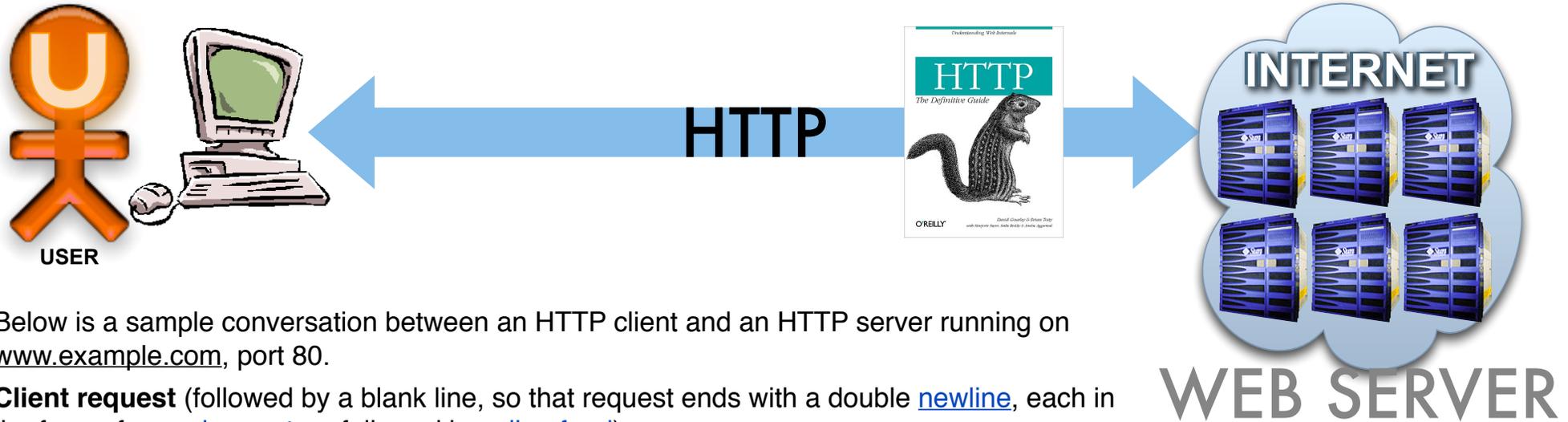


WEB SERVER

Apache, IIS, etc.



# Web Server and HTTP



Below is a sample conversation between an HTTP client and an HTTP server running on [www.example.com](http://www.example.com), port 80.

**Client request** (followed by a blank line, so that request ends with a double [newline](#), each in the form of a [carriage return](#) followed by a [line feed](#)):

```
GET /index.html HTTP/1.1
Host: www.example.com
```

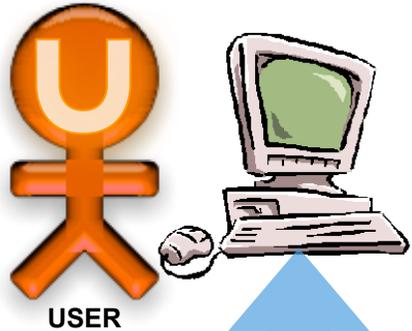
The "Host" header distinguishes between various [DNS](#) names sharing a single [IP address](#), allowing name-based [virtual hosting](#). While optional in HTTP/1.0, it is mandatory in HTTP/1.1.

**Server response** (followed by a blank line and text of the requested page):

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8
```

from <http://en.wikipedia.org/wiki/HTTP>

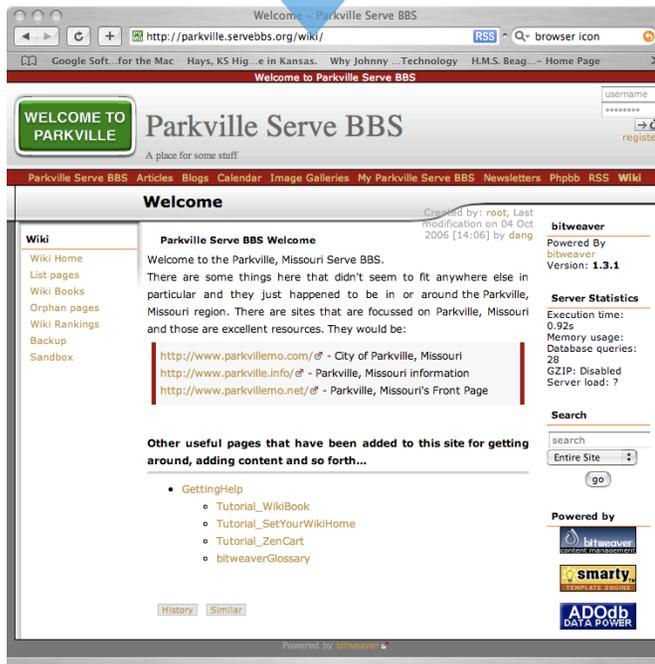
# Web Client & HTML



BROWSER

Internet Explorer,  
Mozilla, Firefox,

Safari,  
etc.



from <http://en.wikipedia.org/wiki/HTML>

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML
2 <html>
3   <head>
4     <title>Example</title>
5     <link href="screen.css" rel="sty
6   </head>
7   <body>
8     <h1>
9       <a href="/">Header</a>
10    </h1>
11    <ul id="nav">
12      <li>
13        <a href="one/">One</a>
14      </li>
15      <li>
16        <a href="two/">Two</a>
17      </li>
```

**- Pick a web site and load it in your browser**  
**- use "File -> View Source" to examine HTML**

# HTML Syntax

from [http://en.wikipedia.org/wiki/http://en.wikipedia.org/wiki/HTML\\_element](http://en.wikipedia.org/wiki/http://en.wikipedia.org/wiki/HTML_element)



from <http://www.w3.org/TR/html401/index/elements.html>

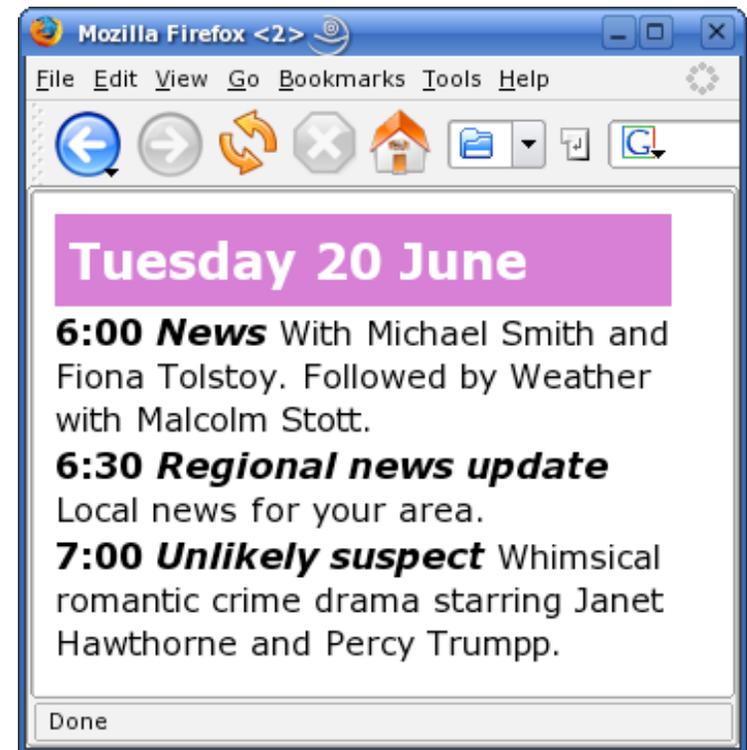
Name	Description
<b>A</b>	<b>anchor</b>
<b>ABBR</b>	<b>abbreviated form (e.g., WWW, HTTP, etc.)</b>
<b>ADDRESS</b>	<b>information on author</b>
<b>APPLET</b>	<b>Java Applet</b>
...	...

# CSS Syntax

from [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)

An XML file containing the following - note the `xml-stylesheet` [processing instruction](#):

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="css.css"?>
<schedule>
  <date>Tuesday 20 June</date>
  <programme>
    <starts>6:00</starts>
    <title>News</title>
    With Michael Smith and Fiona Tolstoy.
    Followed by Weather with Malcolm Stott.
  </programme>
  <programme>
    <starts>6:30</starts>
    <title>Regional news update</title>
    Local news for your area.
  </programme>
  <programme>
    <starts>7:00</starts>
    <title>Unlikely suspect</title>
    Whimsical romantic crime drama starring Janet
    Hawthorne and Percy Trumpp.
  </programme>
</schedule>
```



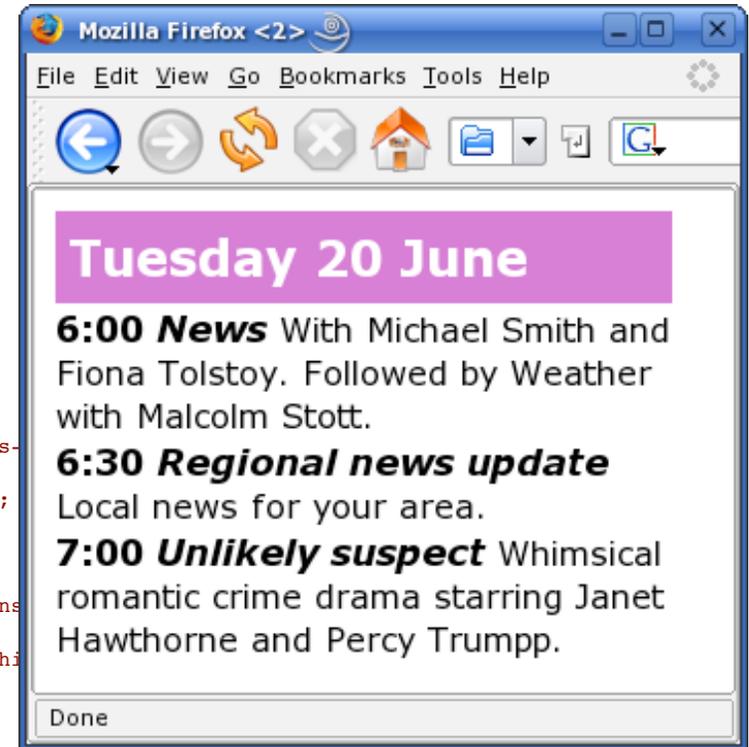
# CSS Example

from [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)

An XML file containing the following - note the `xml-stylesheet` [processing instruction](#):

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="css.css"?>
<schedule>
  <date>Tuesday 20 June</date>
  <programme>
    <starts>6:00</starts>
    <title>News</title>
    With Michael Smith and Fiona Tolstoy.
    Followed by Weather with Malcolm Stott.
  </programme>
  <programme>
    <starts>6:30</starts>
    <title>Regional news update</title>
    Local news for your area.
  </programme>
  <programme>
    <starts>7:00</starts>
    <title>Unlikely suspect</title>
    Whimsical romantic crime drama starring Janet
    Hawthorne and Percy Trumpp.
  </programme>
</schedule>
```

```
@media screen {
  schedule {
    display: block;
    margin: 10px;
    width: 300px;
  }
  date {
    display: block;
    padding: 0.3em;
    font: bold x-large sans-serif;
    color: white;
    background-color: #C6C;
  }
  programme {
    display: block;
    font: normal medium sans-serif;
  }
  programme > * { /* All children
    font-weight: bold;
    font-size: large;
  }
  title {
    font-style: italic;
  }
}
```



# CSS Example

from <http://www.csszengarden.com/>

see also CSS techniques @ <http://meyerweb.com/eric/css/edge/>

css Zen Garden: The Beauty in CSS Design

<http://www.csszengarden.com/> RSS CSS Zen Garden

post to del.icio.us my del.icio.us SJA-winter-...a - cepedia Google Soft...for the Mac Hays, KS Hig...e in Kansas. >>

元  
完  
全  
な  
技

# Zen Garden

The Beauty of CSS Design

*The Road to Enlightenment*

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#)

**So What is This About?**

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what

select a design:

- Retro Theater by Eric Rogé
- Lily Pond by Rose Thorogood
- Icicle Outback by Timo Virtanen
- Zen Army by Carl Desmond
- The Original by Joachim Shotter
- Floral Touch by Jadas Jimmy
- Elegance in Simplicity by Mani Sheriar
- Dazzling Beauty by Deny Sri Supriyono

archives:

[next designs >>](#)

[View All Designs](#)

resources:

[View This Design's CSS](#)

css Zen Garden: The Beauty in CSS Design

<http://www.csszengarden.com/?cssfile=/195/195.css&page=0> RSS CSS Zen Garden

post to del.icio.us my del.icio.us SJA-winter-...a - cepedia Google Soft...for the Mac Hays, KS Hig...e in Kansas. >>

THE BEAUTY OF CSS DESIGN

# cssZenGarden

A DEMONSTRATION OF WHAT CAN BE ACCOMPLISHED VISUALLY THROUGH CSS-BASED DESIGN. SELECT ANY STYLE SHEET FROM THE LIST TO LOAD IT INTO THIS PAGE.

[DOWNLOAD THE SAMPLE HTML FILE AND CSS FILE](#)



 **The Road to Enlightenment**

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the **W3C**, **WaSP** and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

 **SELECT A DESIGN**

-  **Retro Theater** by Eric Rogé
-  **Lily Pond** by Rose Thorogood
-  **Icicle Outback** by Timo Virtanen
-  **Zen Army** by Carl Desmond
-  **The Original** by Joachim Shotter
-  **Floral Touch** by Jadas Jimmy

 **So What is This About?**

# WEB CONTAINERS



The Java plug-in needs your permission to run.

Run this time

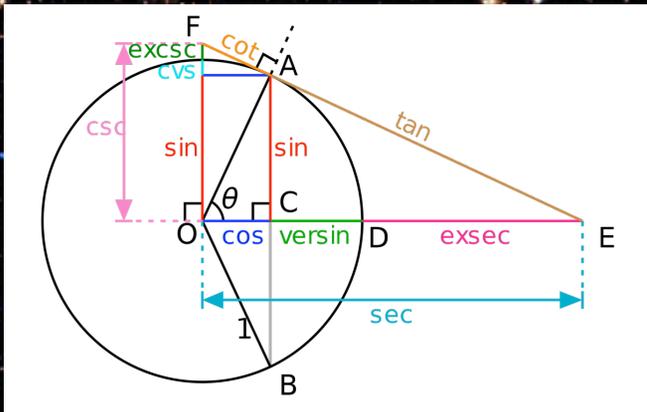
Always run on this site

[Learn more](#) X

```
<html>
<body>
<applet id="ProjectApplet" style="display:block"
code="ScratchApplet" codebase="./"
archive="ScratchApplet.jar" height="387"
width="482">
<param name="project"
value="myscratchproject.sb">
</applet>
</body>
</html>
```

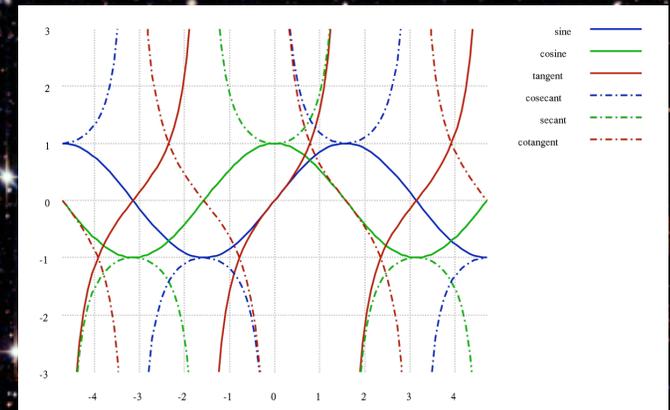


# SCRATCH, MATH & ASTEROIDS



$$x = \frac{\sqrt{4ac + b^2} - b}{2a}$$

$$\cos \theta \equiv \sin \left( \frac{\pi}{2} - \theta \right) \equiv \frac{1}{\sec \theta}$$



“But when are we going to use this?”

# MISSION: Make a Game



- Something the students might already enjoy
  - Involves motion, acceleration, momentum
  - Involves projectiles

<http://scratch.mit.edu/projects/SonicPopsDad/245563>

# CS MISSION: Involve Programming

FEATURES

**Simplest tool**  
Ages 5-15

- Drag and drop code blocks
- 2D graphics frameworks interaction

**Simpler tool**  
Ages 8-22

- Drag and drop code blocks
- 3D graphics frameworks interaction

**Less simple**  
Ages 12-22

- Interactive interpreter code typing
- Media computation helper classes

**Less simple**  
Ages 13-25

- Type, compile, run, debug
- 2D gaming framework interaction via 5 Java classes

**Less simple**  
Ages 15-25

- Type, compile, run, debug
- No default graphics framework environment

**Complex tool**  
Ages 15-25

- Type, compile, run, debug
- No default graphics framework environment

**No/Any tool**  
Ages 16+

- Type, compile, run, debug
- No default graphics framework environment

TOOLS



TEACHING CONCEPTS

- Sequence
- Iteration
- Conditional Logic
- Variables
- Data Structures (dynamic lists)
- Events Handling
- Parallel Execution
- Synchronization
- Random Numbers
- Boolean Logic
- Dynamic Interaction
- User Interface Design
- Publish projects as Java applets on scratch.mit.edu

- All Scratch concepts plus..
- Procedures and Functions
- Parameter Passing & Return Values
- Recursion
- Defining Classes of Objects
- Inheritance
- Text Input

- All previous Alice concepts, excluding user interface design
- Manipulation of audio, images, video through media computation helper classes – see mediacomputation.org
- All Java Programming Language Features Available
- Interpreted mode is great step from drag and drop code blocks (no errors possible) to type – compile – run – debug (all errors possible)

- All previous Alice concepts plus...
- 5 Java classes encapsulate 2D gaming and simulation concepts
- All Java Programming Language Features Available

- Simplest Java IDE
- All Java Programming Language Features Available

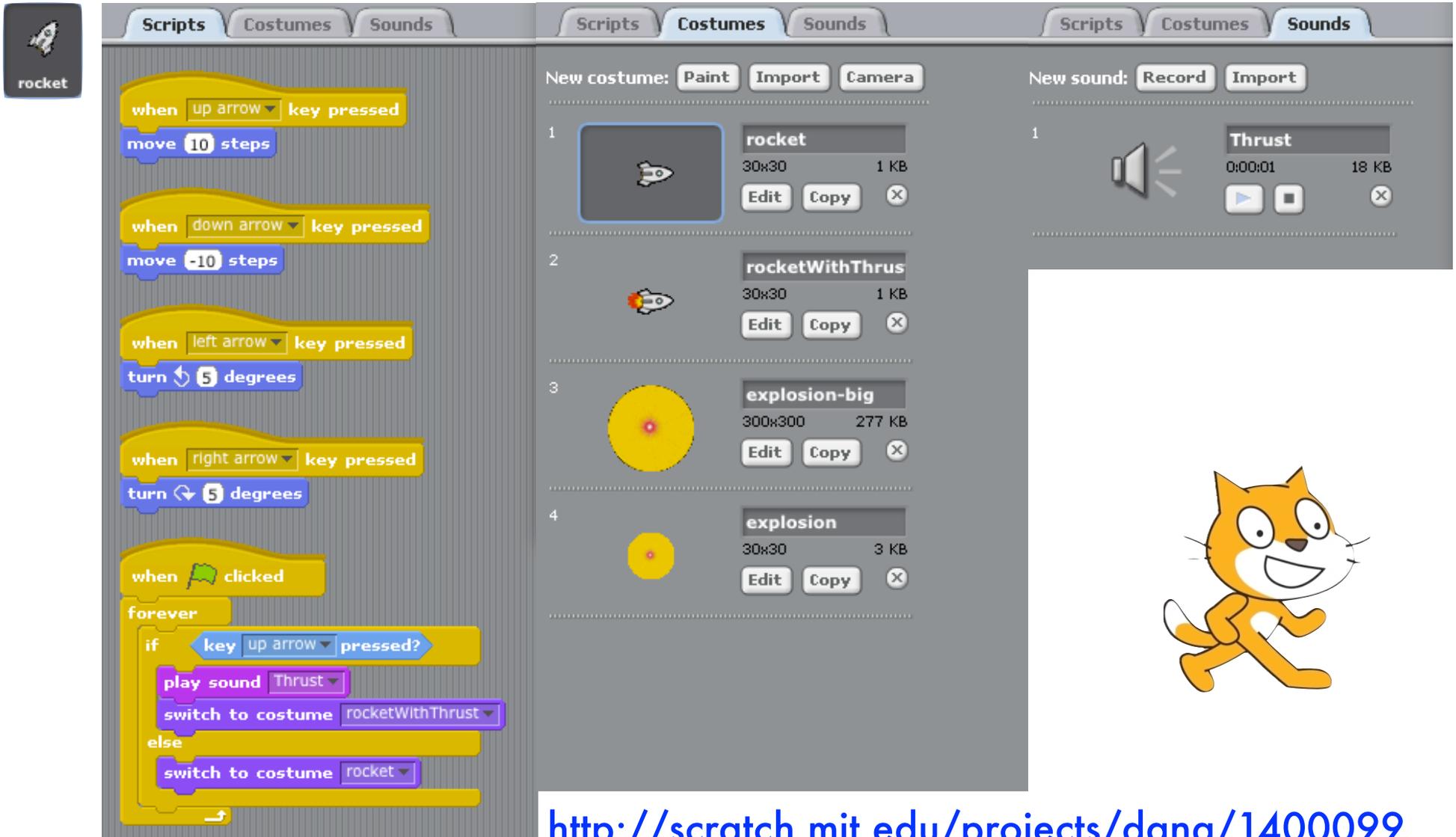
- Complex IDE
- Multiple programming languages available (Java, Ruby, Groovy, Python, PHP, JavaScript, etc.)
- Tooling covers mobile and embedded, enterprise, all avenues

- Programming Language common to Alice, Dr. Java, Greenfoot, BlueJ and NetBeans environments
- All Java Programming Language Features Available
- Tool / environment agnostic

# Scratch Project: Asteroids

Phase	 Scratch	Math	Asteroids
1. Get Moving	Motion, Looks, Sound, Control, Sensing	angle of ship turn, steps to move ship = speed	rocket turn, thrust, movement
2. Off Screen	Motion, Control, Operators	x/y coordinates of ship position	screen wrap around
3. Collisions	Motion, Looks, Sound, Control, Sensing, Operators	variables, conditional logic, event handling	ship crashing into asteroids, asteroids crashing into ship
4. Shooting	Motion, Looks, Sound, Control, Sensing, Operators	message passing, relational expressions	ship shooting bullet, asteroids getting hit by bullet
5. Momentum	Motion, Looks, Control, Sensing, Operators	trigonometric functions, velocity, acceleration	gliding based on momentum and thrust acceleration

# Scratch Phase 1: Get Moving



The image shows the Scratch 2.0 interface with a rocket sprite selected. The Scripts area contains the following code:

- when up arrow key pressed: move 10 steps
- when down arrow key pressed: move -10 steps
- when left arrow key pressed: turn 5 degrees
- when right arrow key pressed: turn 5 degrees
- when green flag clicked: forever loop containing:
  - if key up arrow pressed?: play sound Thrust, switch to costume rocketWithThrust
  - else: switch to costume rocket

The Costumes area shows four costumes:

- rocket (30x30, 1 KB)
- rocketWithThrust (30x30, 1 KB)
- explosion-big (300x300, 277 KB)
- explosion (30x30, 3 KB)

The Sounds area shows one sound:

- Thrust (0:00:01, 18 KB)

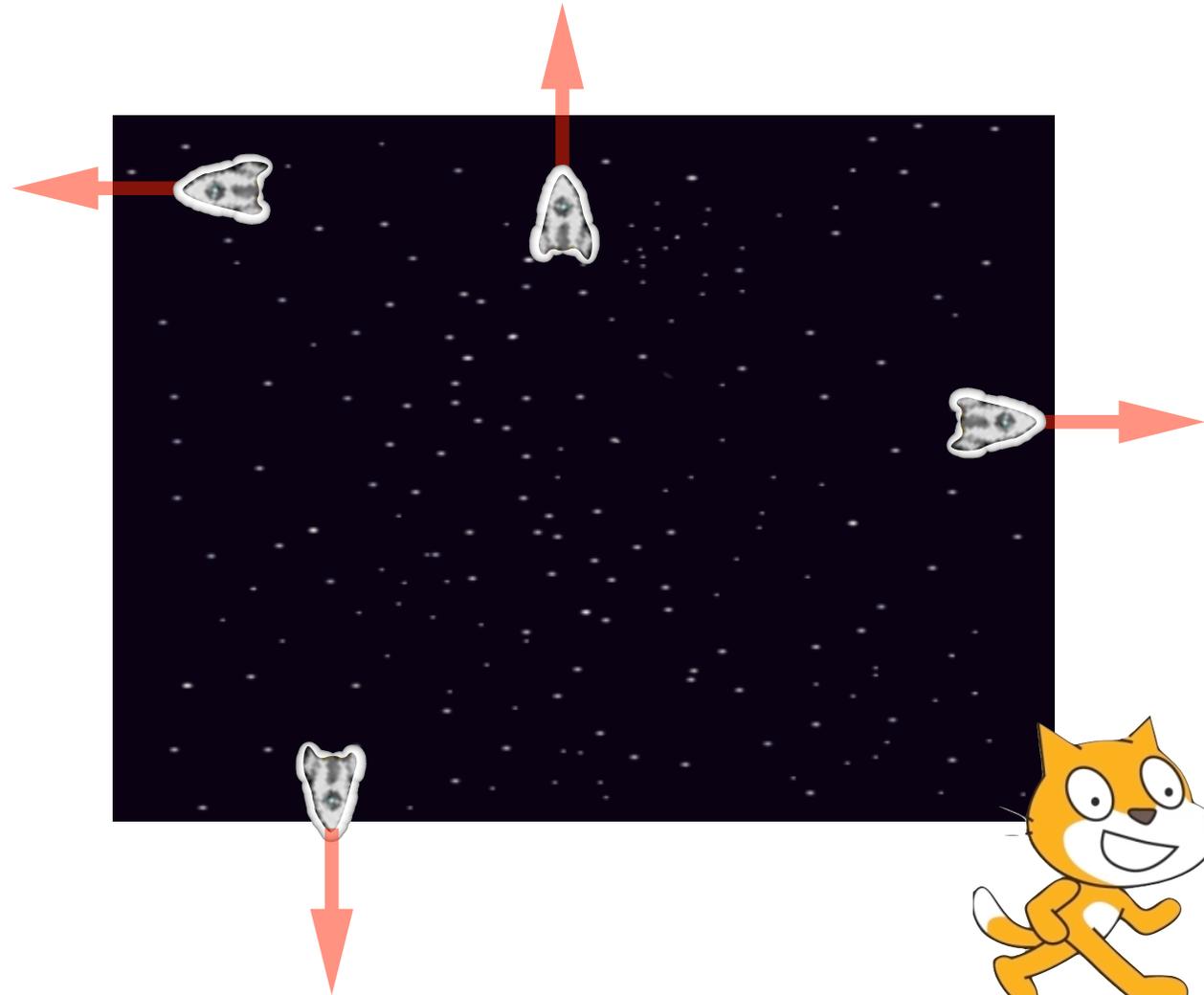
The Scratch logo is visible in the bottom right corner.

<http://scratch.mit.edu/projects/dang/1400099>

# Scratch Phase 2: Off Screen



```
when green flag clicked
forever
  if key up arrow pressed?
    play sound Thrust
    switch to costume rocketWithThrust
  else
    switch to costume rocket
  if x position < -240
    set x to 240
  if x position > 240
    set x to -240
  if y position < -180
    set y to 180
  if y position > 180
    set y to -180
```

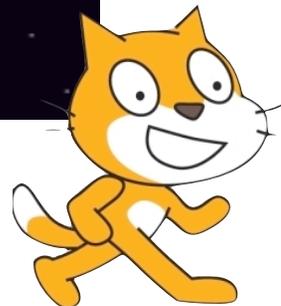
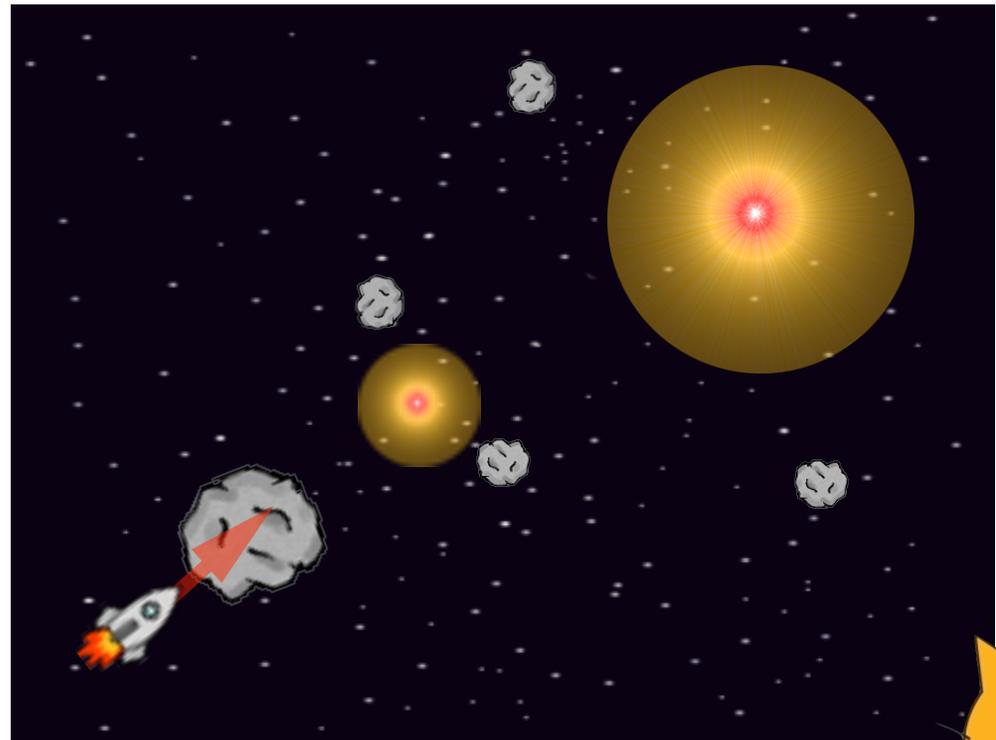


<http://scratch.mit.edu/projects/dang/1400601>

# Scratch Phase 3: Collisions



```
if y position < -180
  set y to 180
if y position > 180
  set y to -180
set crashed to 0
if touching asteroid01 ?
  set crashed to 1
if touching asteroid02 ?
  set crashed to 1
if touching asteroid03 ?
  set crashed to 1
if crashed = 1
  broadcast gameOver
  switch to costume explosion
  wait 0.25 secs
  switch to costume explosion-big
  wait 0.25 secs
  hide
  play sound Screech until done
  stop script
```



<http://scratch.mit.edu/projects/dang/1400765>

# Scratch Phase 3: Collisions

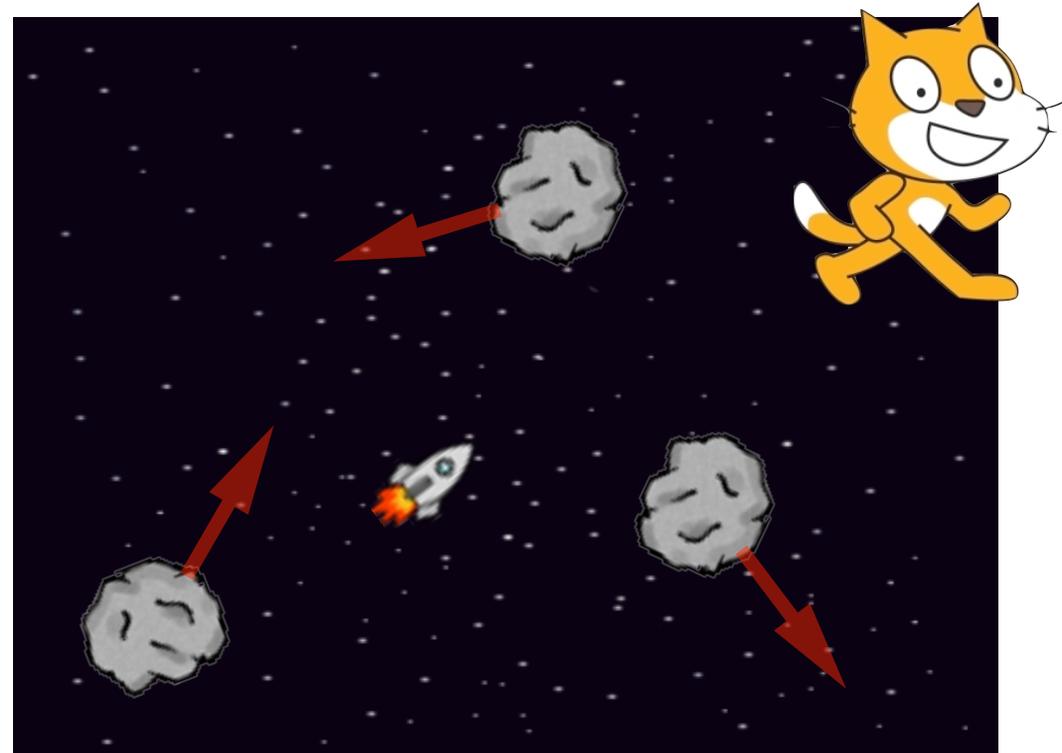


Scratch code blocks for an asteroid's movement and wrap-around logic:

- when clicked
- hide
- go to x: pick random -240 to 240 y: pick random -180 to 180
- point in direction pick random -180 to 180
- show
- forever loop:
  - move 1 steps
- if x position < -240, set x to 240
- if x position > 240, set x to -240
- if y position < -180, set y to 180
- if y position > 180, set y to -180

**move randomly**

**wrap around screen**



<http://scratch.mit.edu/projects/dang/1400765>

- Asteroids are not yet exploding on impact
- They provide something for the rocket to hit

# Scratch Phase 3: Collisions

GAME  
OVER  
gameO...

when clicked

hide

not game  
over yet

when I receive gameOver

show

go to front

forever

change color effect by pick random 1 to 5

change brightness effect by pick random -5 to 5

flicker  
randomly



<http://scratch.mit.edu/projects/dang/1400765>

- **gameOver** message is broadcast from rocket when it collides with an asteroid

# Scratch Phase 4: Shooting

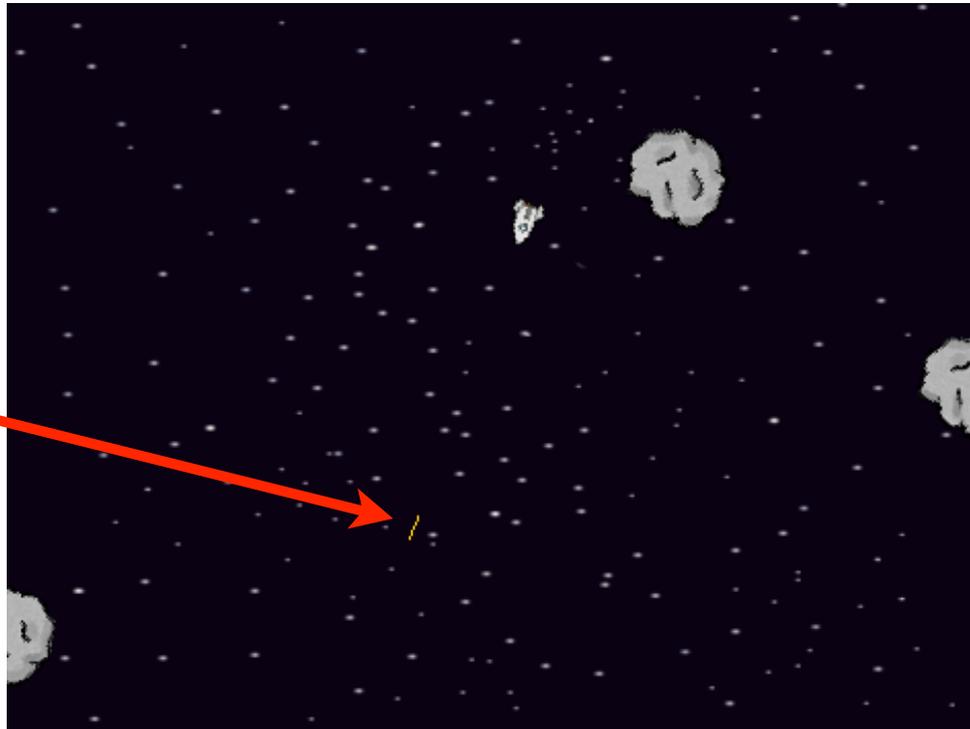


```
when left arrow key pressed
  turn 10 degrees

when right arrow key pressed
  turn 10 degrees

when space key pressed
  play sound Laser1
  broadcast shootBullet01

when clicked
  go to x: 0 y: 0
  show
  forever
    if key up arrow pressed?
      play sound Thrust
      switch to costume rocketWithThrust
    else
      switch to costume rocket
  if x position < -240
    set x to 240
```



<http://scratch.mit.edu/projects/dang/1402543>

- **shootBullet01** message is broadcast from rocket when space key is pressed

# Scratch Phase 4: Shooting

when clicked

hide

**don't show up until  
there's shooting**

when I receive shootBullet01

go to x: x position of rocket y: y position of rocket

point in direction direction of rocket

show

**shoot from  
rocket**

repeat 20

move 10 steps

if x position < -240

set x to 240

if x position > 240

set x to -240

if y position < -180

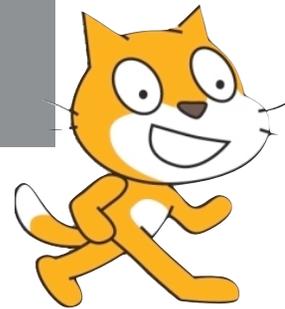
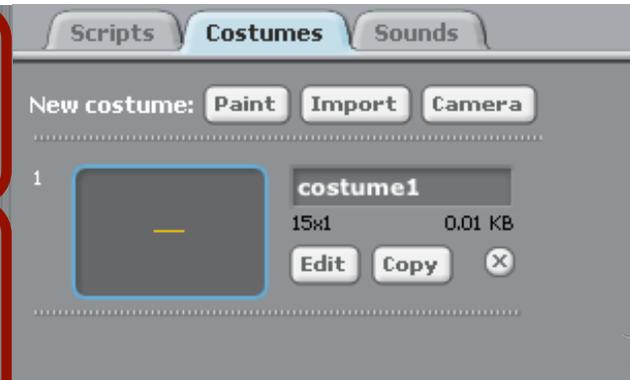
set y to 180

if y position > 180

set y to -180

hide

**shoot forward  
from rocket and  
wrap around  
screen**



<http://scratch.mit.edu/projects/dang/1402543>

- This version only deals with shooting one bullet at a time
- Students can add **bullet02**, **bullet03**, and scripting

# Scratch Phase 4: Shooting



**big**  
**1**

```
if touching rocket ? or touching bullet01 ?  
  broadcast asteroid01Explodes  
  play sound Pop until done  
  hide  
  stop script
```



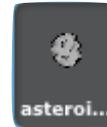
**big**  
**2**

```
if touching rocket ? or touching bullet01 ?  
  broadcast asteroid02Explodes  
  play sound Pop until done  
  hide  
  stop script
```



**big**  
**3**

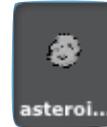
```
if touching rocket ? or touching bullet01 ?  
  broadcast asteroid03Explodes  
  play sound Pop until done  
  hide  
  stop script
```



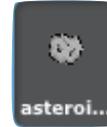
**small**  
**1**



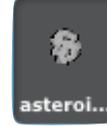
**small**  
**2**



**small**  
**3**



**small**  
**4**



**small**  
**5**



**small**  
**6**

- Listening for **asteroid01Explodes**
- Listening for **asteroid01Explodes**
- Listening for **asteroid02Explodes**
- Listening for **asteroid02Explodes**
- Listening for **asteroid03Explodes**
- Listening for **asteroid03Explodes**



# Scratch Phase 4: Shooting



small  
1

```
when I receive asteroid01Explodes
  go to x: x position of asteroid01 y: y position of asteroid01
  point in direction direction of asteroid01 + pick random -45 to 45
  show

forever
  move 1 steps
  if x position < -240
    set x to 240
  if x position > 240
    set x to -240
  if y position < -180
    set y to 180
  if y position > 180
    set y to -180

if touching rocket? or touching bullet01?
  play sound Pop until done
  hide
  stop script
```

wrap  
around  
screen



**appear near where the bigger asteroid exploded traveling in roughly same direction**

- Each smaller asteroid belongs to a larger asteroid
- Each smaller asteroid sprite has the same script except for responding to **asteroid01**, **asteroid02**, or **asteroid03**

**if hit by rocket or bullet, explode and don't worry about smaller chunks**

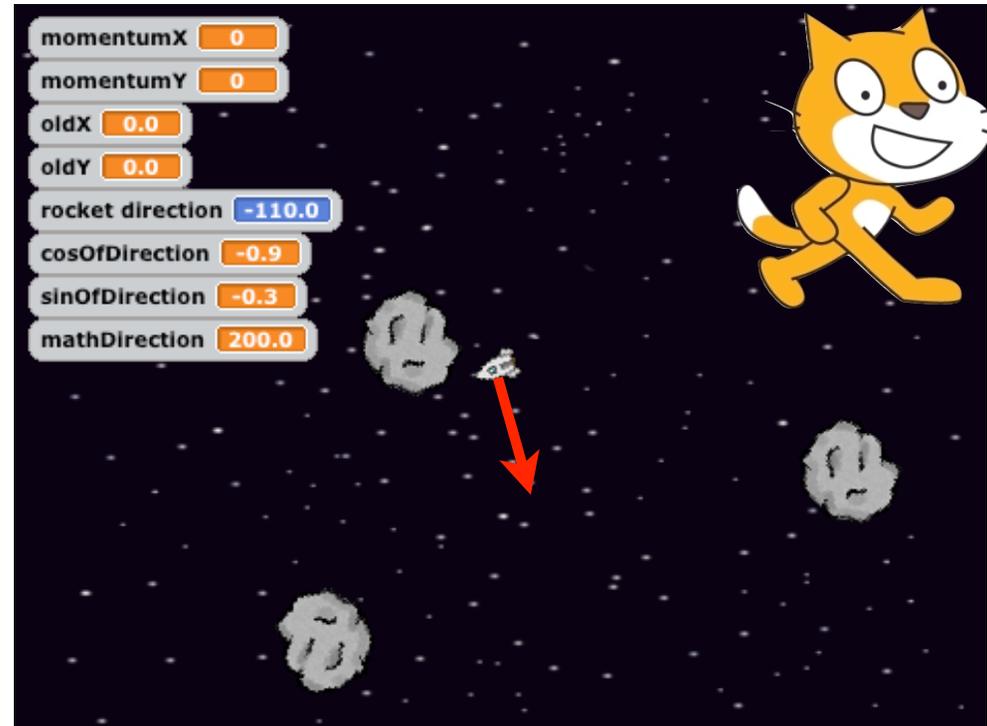
# Scratch Phase 5: Momentum



```
when up arrow key pressed
  set mathDirection to 90 - direction
  change momentumX by cos of mathDirection * 0.25
  change momentumY by sin of mathDirection * 0.25

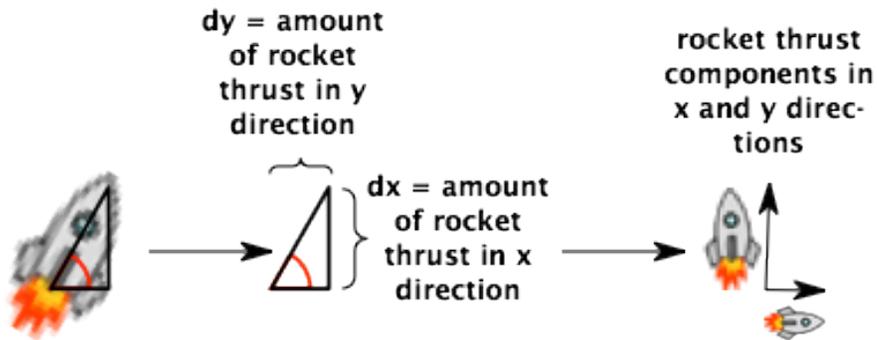
when clicked
  forever
    set mathDirection to 90 - direction
    set sinOfDirection to sin of mathDirection
    set cosOfDirection to cos of mathDirection

when clicked
  set momentumX to 0
  set momentumY to 0
  go to x: 0 y: 0
  show
  forever
    set oldX to x position
    set oldY to y position
    set x to x position + momentumX
    set y to y position + momentumY
```



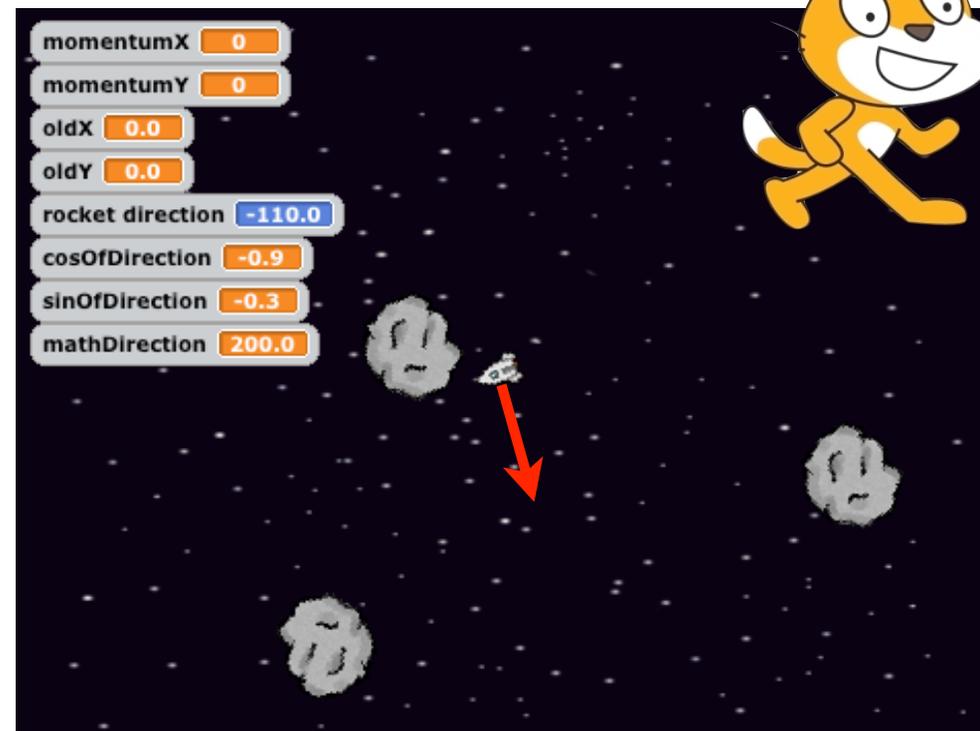
- Major switch in moving the ship from stop and go to maintaining directional momentum
  - Adjusted **mathDirection** from Scratch built in **direction** variable
- <http://scratch.mit.edu/projects/dang/1423848>

# Momentum - Part 1



- In order to keep your rocket ship from the Asteroids game flying around in the right direction and maintaining momentum, there are **two things** you'll need to do get that working

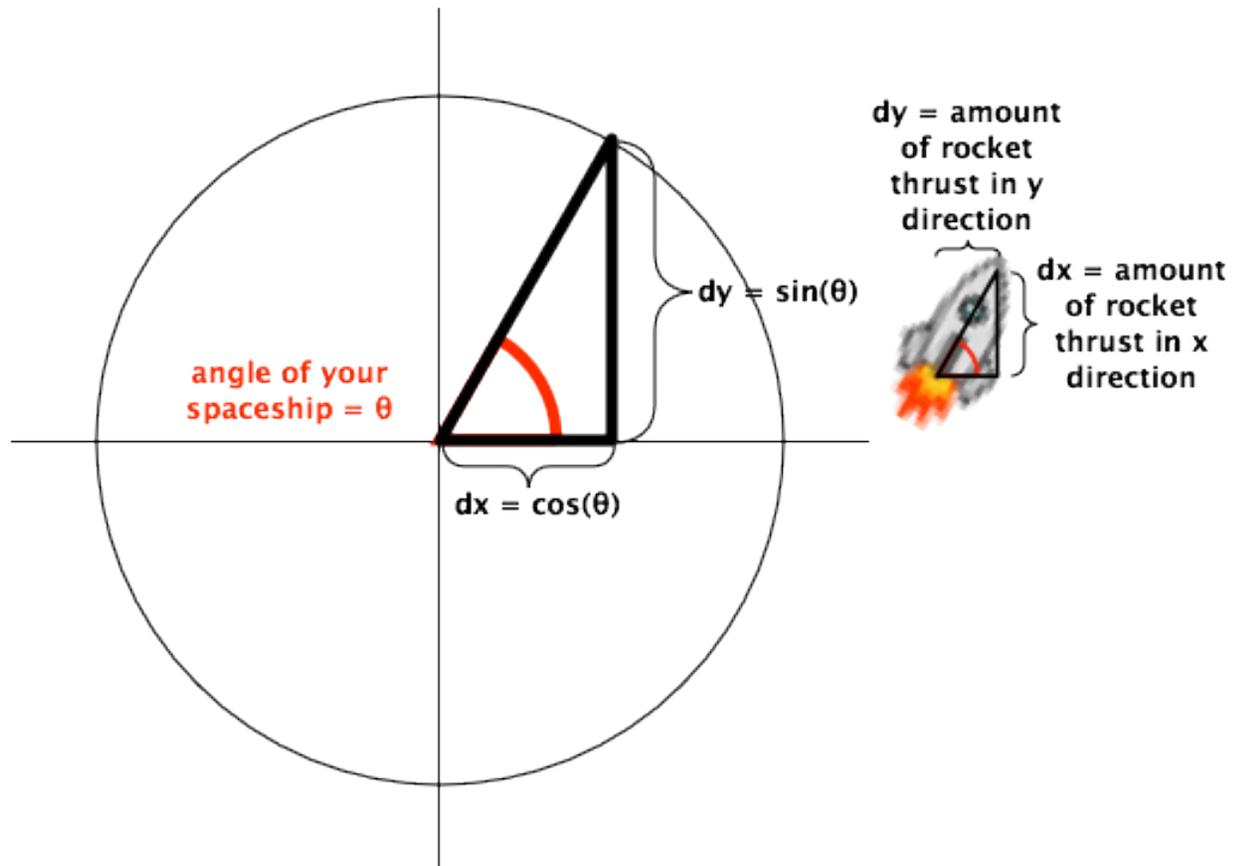
- **THING 1** - We need to know how big or small of change in both the x and y direction to apply as you zoom around the screen with your rocket turned on



<http://scratch.mit.edu/projects/dang/1423848>

# Momentum - Part 2

● **THING 2** - We need to know how big or small of a change in momentum. If you liken this triangle of rocket tilt to a circle, you can imagine your rocket is the radius of the circle and for the purposes of telling Scratch how much or little to move us along the x axis and y axis, we're interested in getting the x and y component of the triangle formed at every position around the circle as the rocket is rotating as illustrated in the figure here

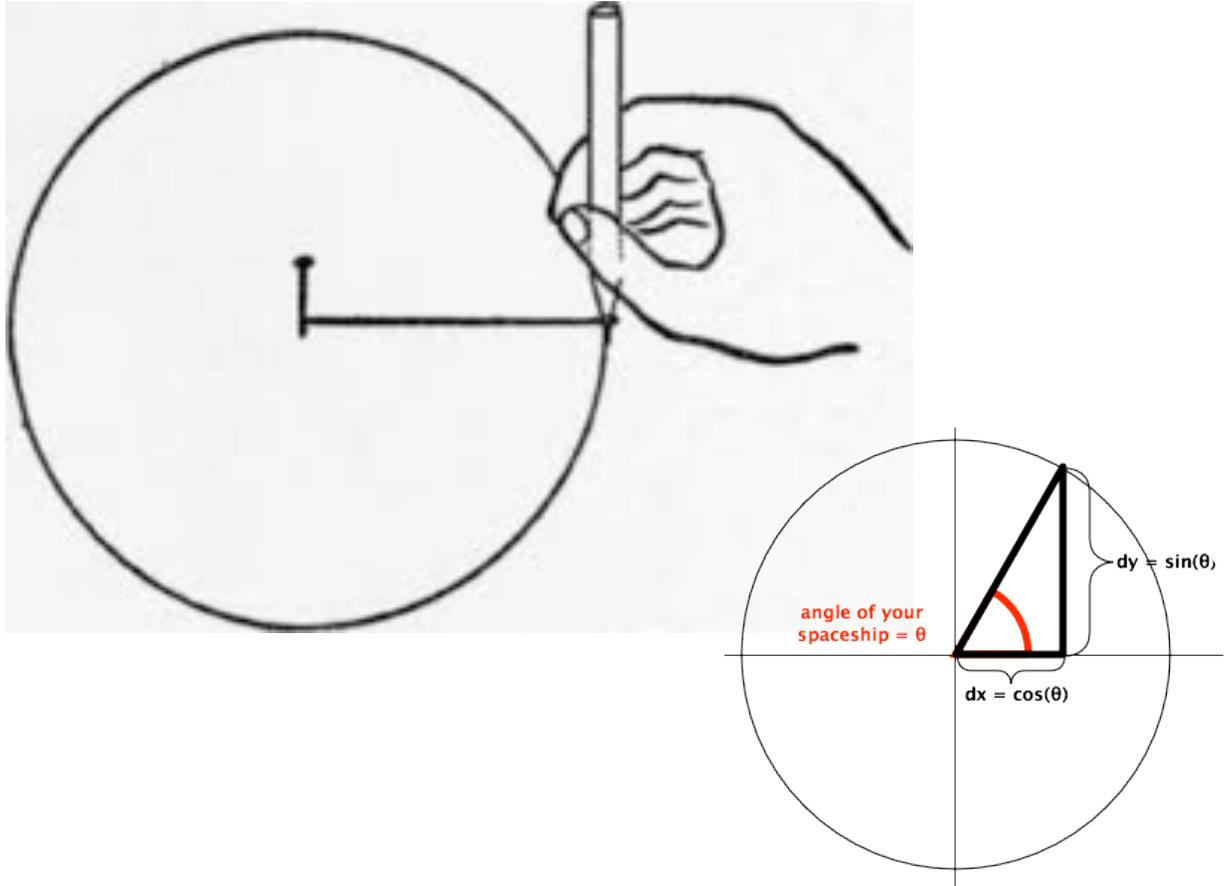


# Momentum - Part 3

- **TRIG FUNCTIONS** -

Thankfully there are handy trigonometric functions available to us that give us exactly this, the x and y component of a triangle in this way. You may have already hit this in school, known as the formula for a circle, where, placing a circle with origin at  $x=0$ ,  $y=0$ , all points along the

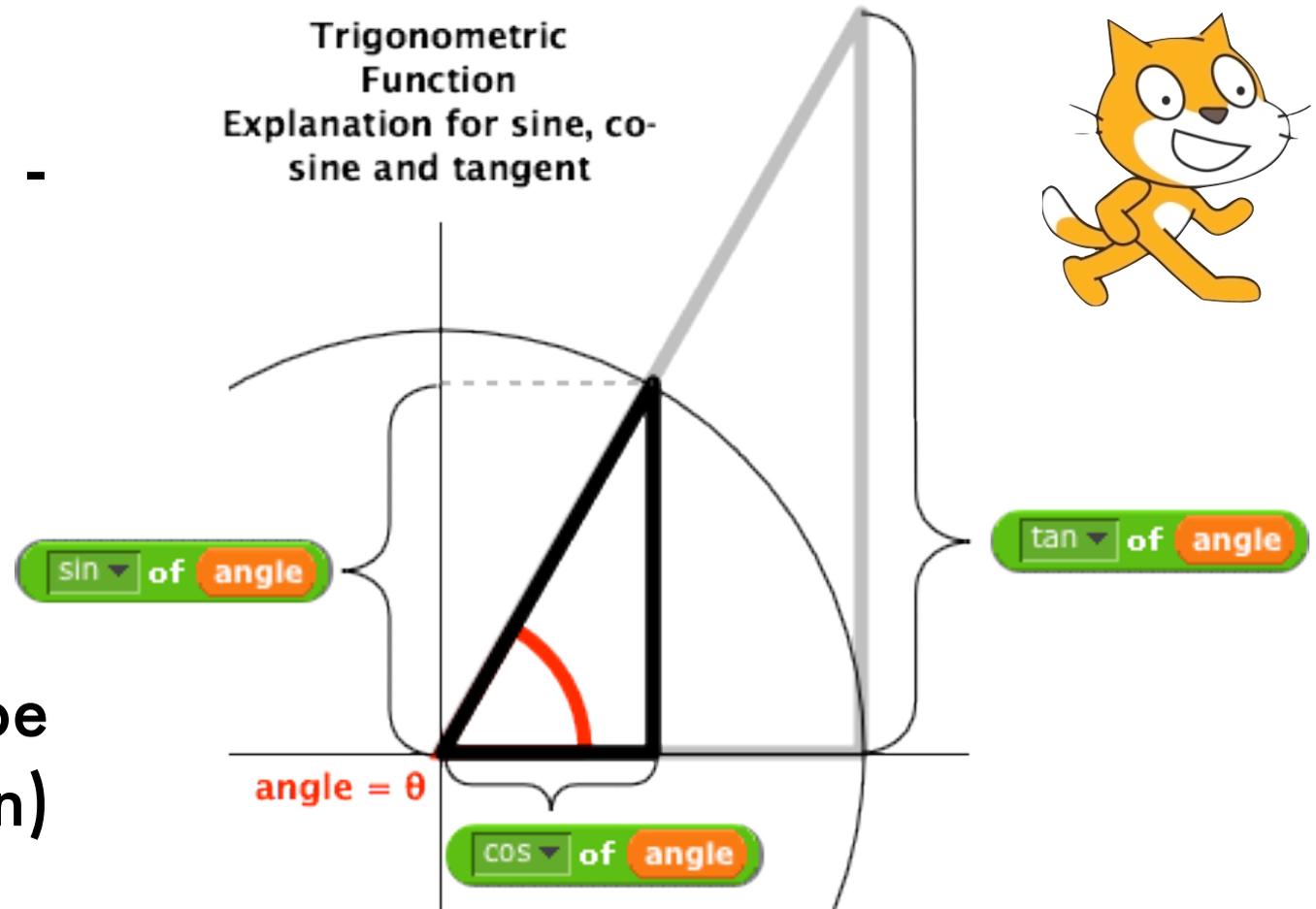
circle can be described as the radius squared is equal to the sum of the x position squared plus the y position squared, or  $r^2 = x^2 + y^2$ . An easy way to think about this is drawing a circle with a pencil, a piece of string, and a pin, it would look like the figure above



# Momentum - Part 4

- **TRIG & SCRATCH** -

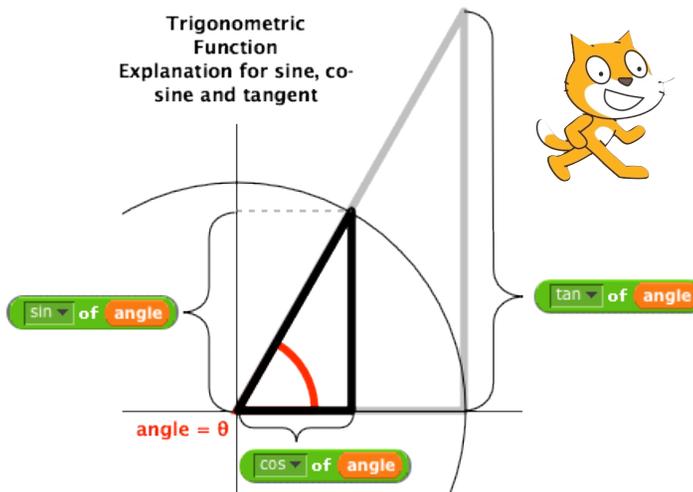
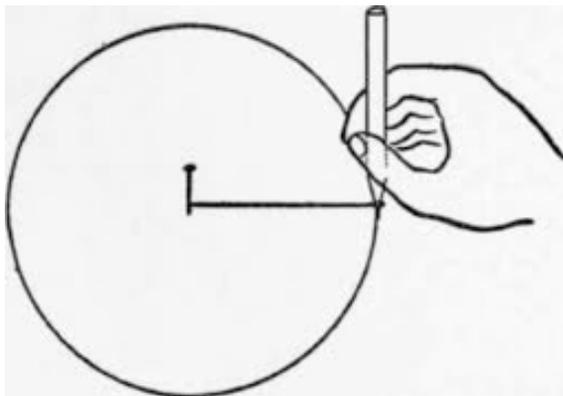
The relationship of functions we'll use and which Scratch blocks you'll need are illustrated here, note that we won't be needing tangent (tan) but it's here for completeness, almost any discussion including sine and cosine will also mention tangent.



# Momentum - Part 5

- **FROM ZERO TO ONE AND BACK**

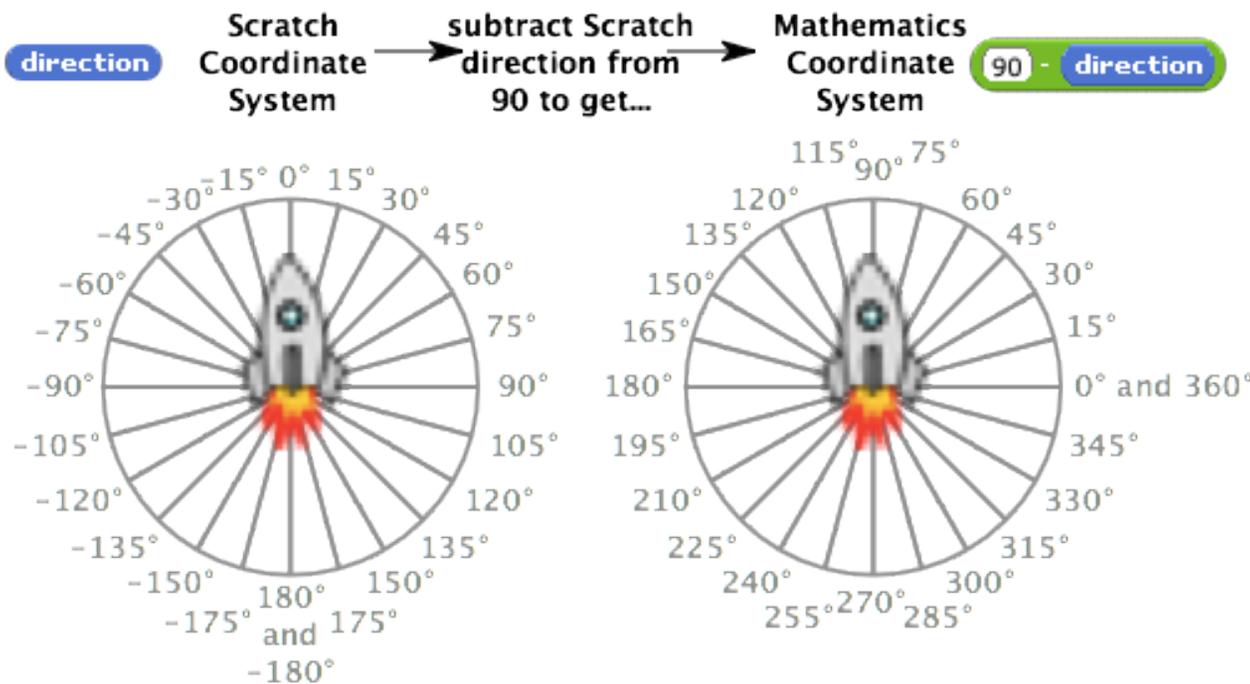
**AGAIN** - Rather than draw lots of different triangles representing you spinning your rocket ship around, here are the values these functions return given the angle you put in, the table below shows values for a few different angles. For our purposes, you'll be using the built in `direction` variable in scratch for your angle.



angle= $\theta$	cos( $\theta$ )	sin( $\theta$ )	tan( $\theta$ )
0	1.0	0.0	0.0
15	0.9659	0.2588	0.2679
30	0.8660	0.5	0.5773
45	0.7071	0.7071	1.0
60	0.5	0.8660	1.7320
75	0.2588	0.9659	3.7320
90	0.0	1.0	infinity
105	-0.2588	0.9659	-3.7320
120	-0.5	0.8660	-1.7320
135	-0.7071	0.7071	-1.0
150	-0.8660	0.5	-0.5773
165	-0.9659	0.2588	-0.2679
180	-1.0	0.0	0.0
195	-0.9659	-0.2588	0.2679
210	-0.8660	-0.5	0.5773
225	-0.7071	-0.7071	1.0
240	-0.5	-0.8660	1.7320
255	-0.2588	-0.9659	3.7320
270	0.0	-1.0	infinity
285	0.2588	-0.9659	-3.7320
300	0.5	-0.8660	-1.7320
315	0.7071	-0.7071	-1.0
330	0.8660	-0.5	-0.5773
345	0.9659	-0.2588	-0.2679
360	1.0	0.0	0.0

# Momentum - Part 6

- **BUT IT DOESN'T WORK???** - If you're following along so far and have tried all this, you'll notice it doesn't work correctly, the ship flies off in directions you don't expect. Well, here the reason: the coordinate system for direction used by scratch is not the same as the coordinate system we typically see in math examples, namely, most math examples assume angle=0 points to the right and Scratch assumes angle=0 points up.



up. Neither is right or wrong, you can spin your coordinate system any way you'd like. And that's exactly what we'll do as shown in the diagram here.

# Momentum - Part 7



```
when up arrow key pressed
set mathDirection to 90 - direction
change momentumX by cos of mathDirection * 0.25
change momentumY by sin of mathDirection * 0.25
```

- **MOVE** - repeatedly move ship around based on momentum in the x and y direction

```
when clicked
set momentumX to 0
set momentumY to 0
go to x: 0 y: 0
show
forever
set oldX to x position
set oldY to y position
set x to x position + momentumX
set y to y position + momentumY
```