CS74.42A Game Dev 1

Fall 2017 ~ Ethan Wilde

Week 2

SANTA ROSA
JUNIOR COLLEGE

Course Outline

1 World of Game Development / Introduction to JavaScript	10 Modeling and Rigging / Learning Physics
2 Get Started with Browser-Based 2D Games / More JavaScript	11 Advanced Physics and Special Effects
3 Build from a Blueprint: Game Design Document	12 Final Project: Design Your Game
4 Build and Playtest Sprint 1: Midterm Project	13 Build and Playtest Sprint 1: Final Project
5 Build and Playtest Sprint 2: Midterm Project	14 Enhancing UI and Sound
6 Midterm Project Review	15 Build and Playtest Sprint 2: Final Project
7 Introducing Unity: Building Games for Multiple Platforms	16 Finalizing and Optimizing
8 Create a Scene	17 Taking Your Game Further
9 Start Scripting / Midterm Exam	18 Final Exam / Final Project Review Discussion

Get all of the details in the complete syllabus on Canvas.

Introduction to JavaScript, Part 2

> **“It became clear around 2010 or so that JS and HTML would be the future of interactive content on the web.”**
>
> — *Travis Faas*

JavaScript Basics

1. Instructions (Lexical Structures)
2. Comments
3. Values + Variables
4. Expressions + Operators
5. Statements + Control Structures
6. Functions
7. JavaScript, the Web + Cloud9

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

Instructions
(Lexical Structures)

break
case
catch
continue
debugger
default
delete
do
else
false
finally
for
function
if
in
instanceof
new
null
return
switch
this
throw
true
try
typeof
var
void
while
with

RESERVED WORDS

Instructions (Lexical Structures)

```
let a = 0;
```

```
let a = 0
```

END OF LINE (OPTIONAL)

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

Comments

```
// a single line comment

/*
a multi-line
comment
*/
```

COMMENTS

JavaScript Basics

1. Instructions (Lexical Structures)
2. Comments
3. Values + Variables
4. Expressions + Operators
5. Statements + Control Structures
6. Functions
7. JavaScript, the Web + Cloud9

Values + Variables

Number

String

Boolean

Null

Undefined

Symbol

PRIMITIVE TYPES AND OBJECTS

Values + Variables

Number
String
Boolean

Null
Undefined

Symbol

Array Objects

Object Values

PRIMITIVE TYPES AND OBJECTS

Values + Variables

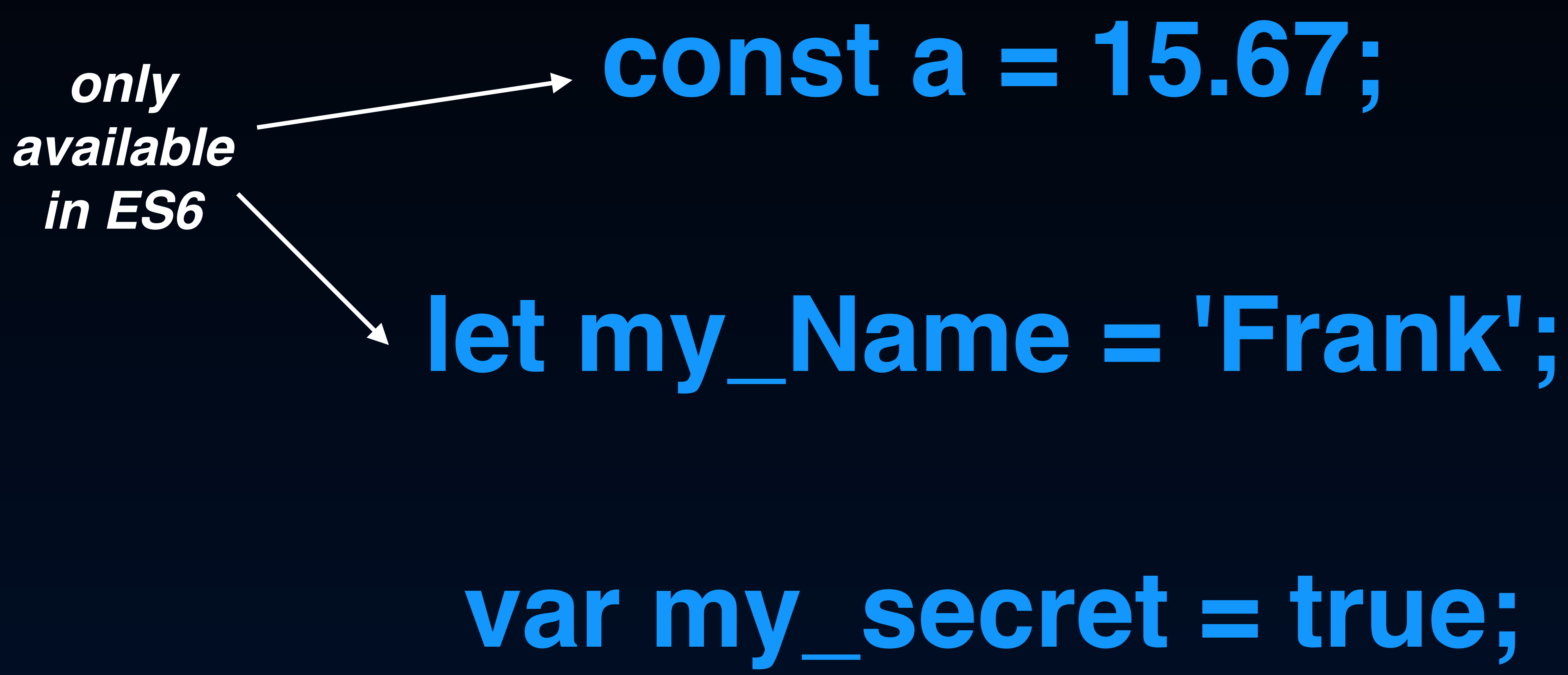
```
const a = 15.67;

let my_Name = 'Frank';

var my_secret = true;
```

NUMBERS, STRINGS + BOOLEANS

Values + Variables



only available in ES6

```
const a = 15.67;

let my_Name = 'Frank';

var my_secret = true;
```

NUMBERS, STRINGS + BOOLEANS

Values + Variables

```
let a = Math.round( 0.6 );

let b = Math.random( );
```

MATH OBJECT FOR ARITHMETIC

Values + Variables

```
let q = [ 'Zero', 'One', 'Two' ];
```

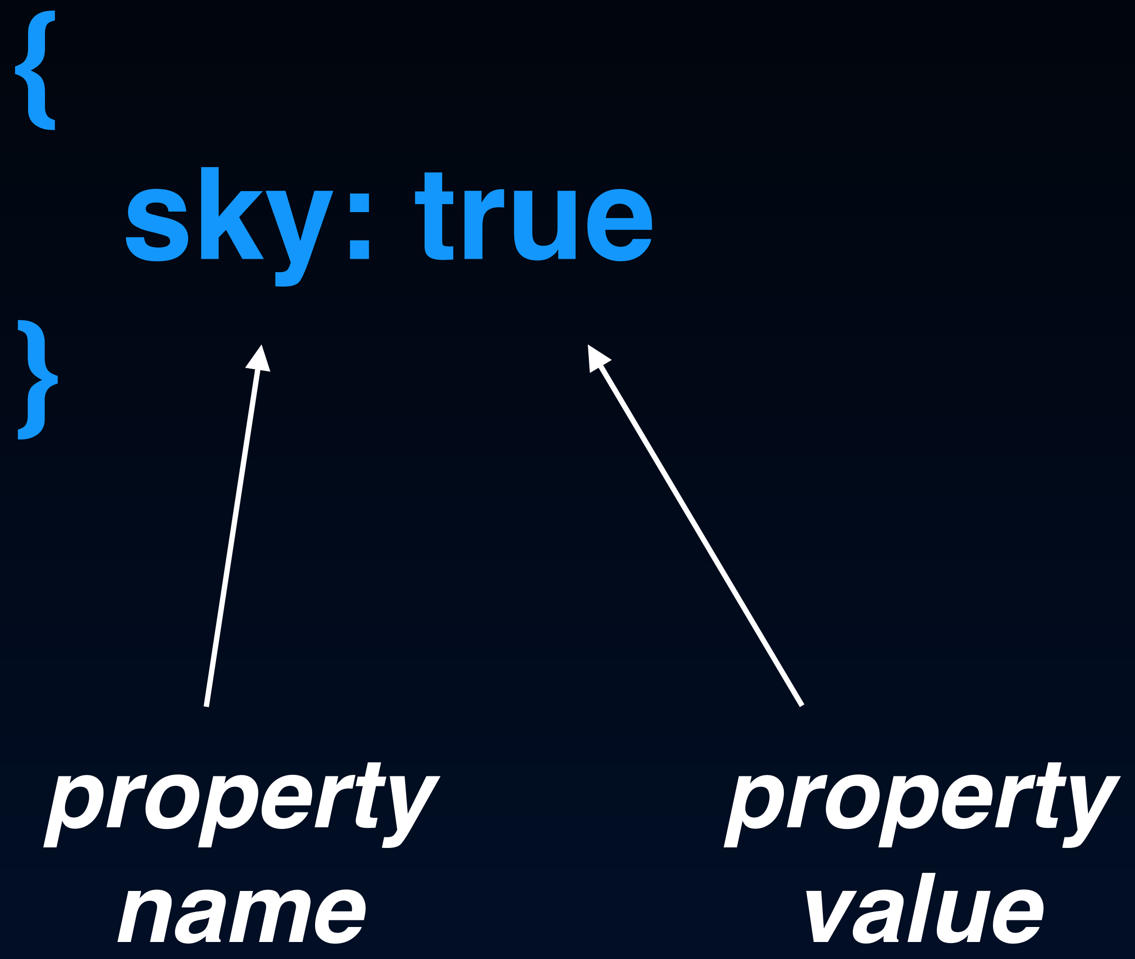
ARRAYS A.K.A. LISTS OF VALUES

Values + Variables

```
{
  sky: true
}
```

OBJECT VALUES

Values + Variables



```
{
  sky: true
}
```

property name

property value

OBJECT VALUES

Values + Variables

```
let world = {
   sky: true
};
```

OBJECT VALUES

Values + Variables

```
let world = {
    sky: true,
    land: true
};
```

OBJECT VALUES

Values + Variables

```
let world = {
        sky: true,
        land: true,
        hours: 24,
        sea : {
            color: "blue",
            actions: {
                wave: function() {
                    // do something
                }
            },
            temp: 72.5
        }
}
```

property with object value

OBJECT VALUES

Values + Variables

```
let world = {
        sky: true,
        land: true,
        hours: 24,
        sea : {
            color: "blue",
            actions: {
                wave: function() {
                    // do something
                }
            },
            temp: 72.5
        }
}
```

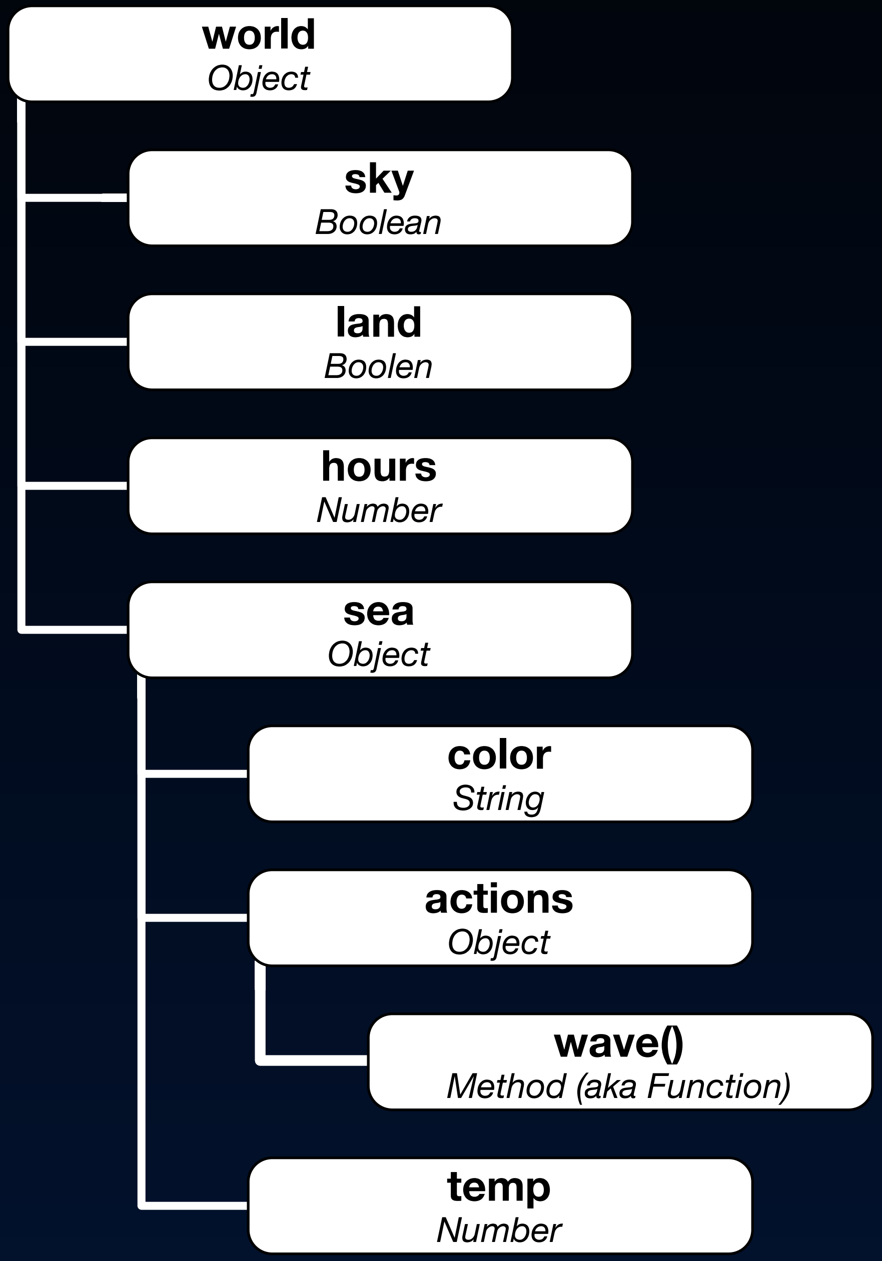
property with object value

property with function value called method

OBJECT VALUES

Values + Variables

```
let world = {
    sky: true,
    land: true,
    hours: 24,
    sea : {
        color: "blue",
        actions: {
            wave: function() {
                // do something
            }
        },
        temp: 72.5
    }
}
```



OBJECT VALUES AS STRUCTURES

Values + Variables

```
let world = {
        sky: true,
        land: true,
        hours: 24,
        sea : {
            color: "blue",
            actions: {
                wave: function() {
                    // do something
                }
            },
            temp: 72.5
        }
}
```

world.sky

world.sea.color

world.sea.actions.wave()

OBJECT VALUES + DOT NOTATION

JavaScript Basics

1. Instructions (Lexical Structures)
2. Comments
3. Values + Variables
4. **Expressions + Operators**
5. Statements + Control Structures
6. Functions
7. JavaScript, the Web + Cloud9

Expressions + Operators

```
let a = ( 4 + 6 ) * 200;
```

EXPRESSIONS EVALUATE TO A VALUE

Expressions + Operators

+ +	increment	===	equals
- -	decrement	!==	not equals
-	subtract	<	less than
+	add or concat	>	greater than
*	multiply	<=	lt or equal
/	divide	>=	gt or equal
!	not	=	assignment

COMMON OPERATORS

JavaScript Basics

1. Instructions (Lexical Structures)
2. Comments
3. Values + Variables
4. Expressions + Operators
5. Statements + Control Structures
6. Functions
7. JavaScript, the Web + Cloud9

Statements + Control Structures

only available in ES6

```
var y = 965;
let x = 100.25;
const z = 12;

function my_guy()  {
    // code block
}
```

DECLARATION STATEMENTS

Statements +
Control Structures

```
if ( x === 1 )  {
    // block of code if true
} else {
    // block of code if false
}
```

CONDITIONAL STATEMENTS

Statements + Control Structures

```
switch ( x )  {
    case 1:
        // do if x is 1
        break;
    default:
        // do if nothing else
}
```

CONDITIONAL STATEMENTS

Statements + Control Structures

```
while ( c < 10 ) {
      c = c + 2;
}
```

LOOPS

Statements + Control Structures

```
var i;
for ( i = 1;  i <= 10;  i++ ) {
     alert( i );
}
```

LOOPS

JavaScript Basics

1. Instructions (Lexical Structures)
2. Comments
3. Values + Variables
4. Expressions + Operators
5. Statements + Control Structures
6. **Functions**
7. JavaScript, the Web + Cloud9

Functions

```
function sneeze( x )  {
    let y;
    if ( x < 100 ) {
        y = x * 2;
    } else {
        y = x * 3;
    }
    return y;
}
```

ACCEPT PARAMETERS + RETURN VALUES

Functions

```
alert ( sneeze( 50 ) );
```

```
function sneeze( x )  {
    let y;
    if ( x < 100 ) {
        y = x * 2;
    } else {
        y = x * 3;
    }
    return y;
}
```

WHAT NUMBER WILL BE DISPLAYED?

JavaScript Basics

1. **Instructions (Lexical Structures)**
2. **Comments**
3. **Values + Variables**
4. **Expressions + Operators**
5. **Statements + Control Structures**
6. **Functions**
7. **JavaScript, the Web + Cloud9**

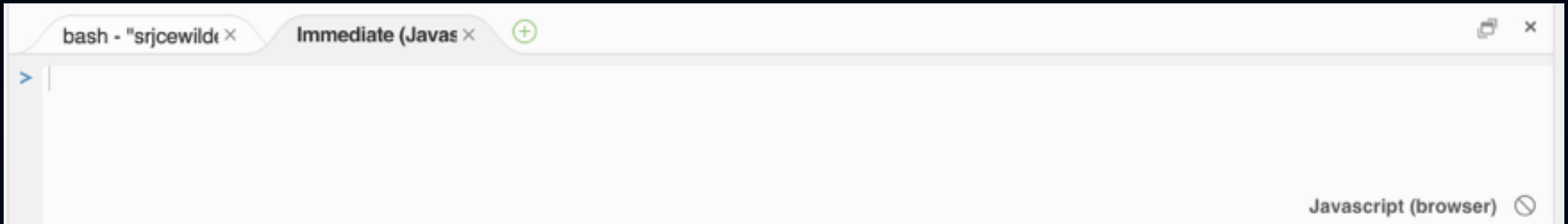
JavaScript in Web Browsers

```
document.write( 'Frank' );

window.location.href =
'http://santarosa.edu';
```

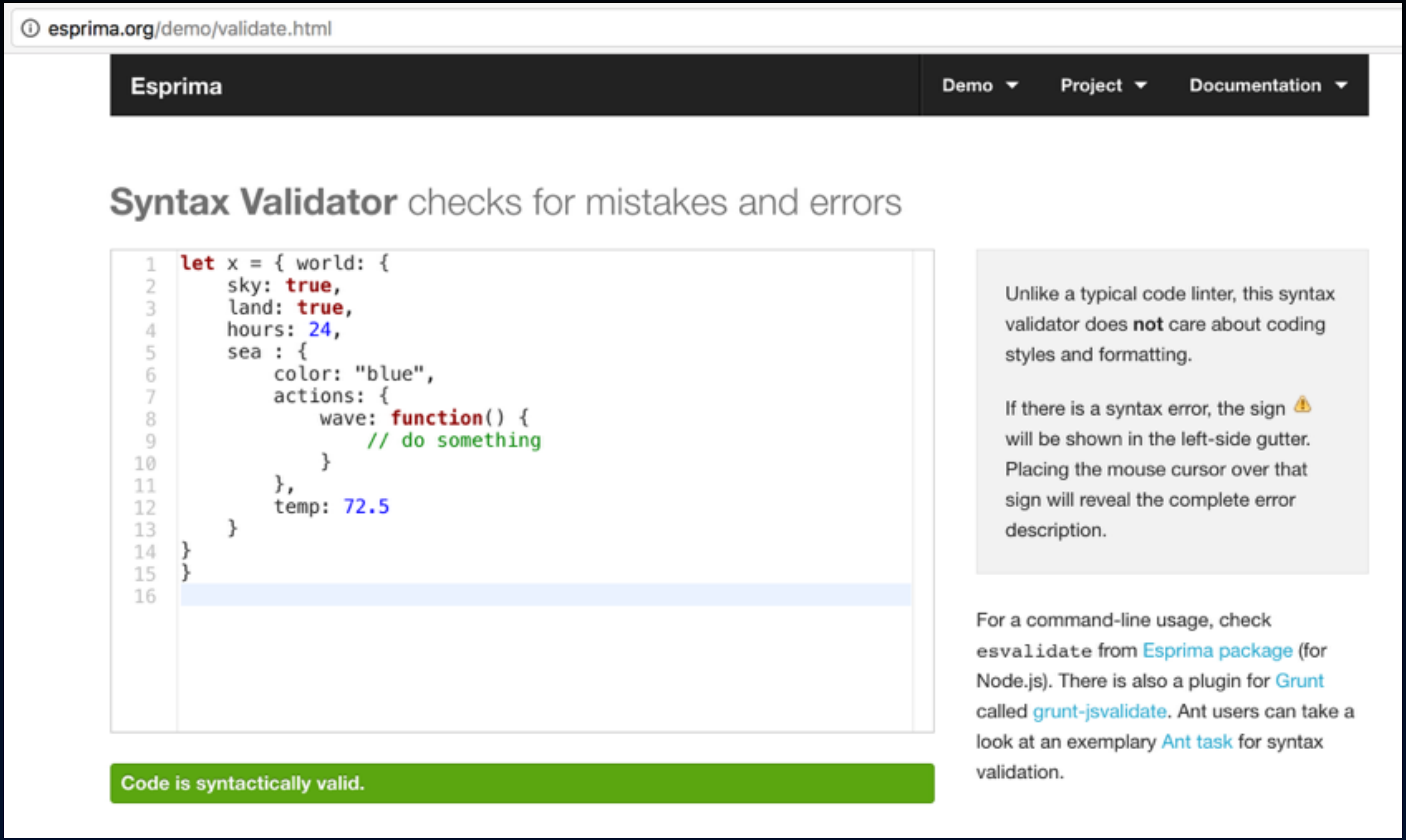
DOT NOTATION + THE OBJECT MODEL

JavaScript Coding in Cloud9



Cloud9 offers a built-in JavaScript console, where we can test out code in real time.

Validating JavaScript Code



Syntax Validator checks for mistakes and errors

```
let x = { world: {
    sky: true,
    land: true,
    hours: 24,
    sea : {
        color: "blue",
        actions: {
            wave: function() {
                // do something
            }
        },
        temp: 72.5
    }
}
}

```

Code is syntactically valid.

Unlike a typical code linter, this syntax validator does **not** care about coding styles and formatting.

If there is a syntax error, the sign ⚠ will be shown in the left-side gutter. Placing the mouse cursor over that sign will reveal the complete error description.

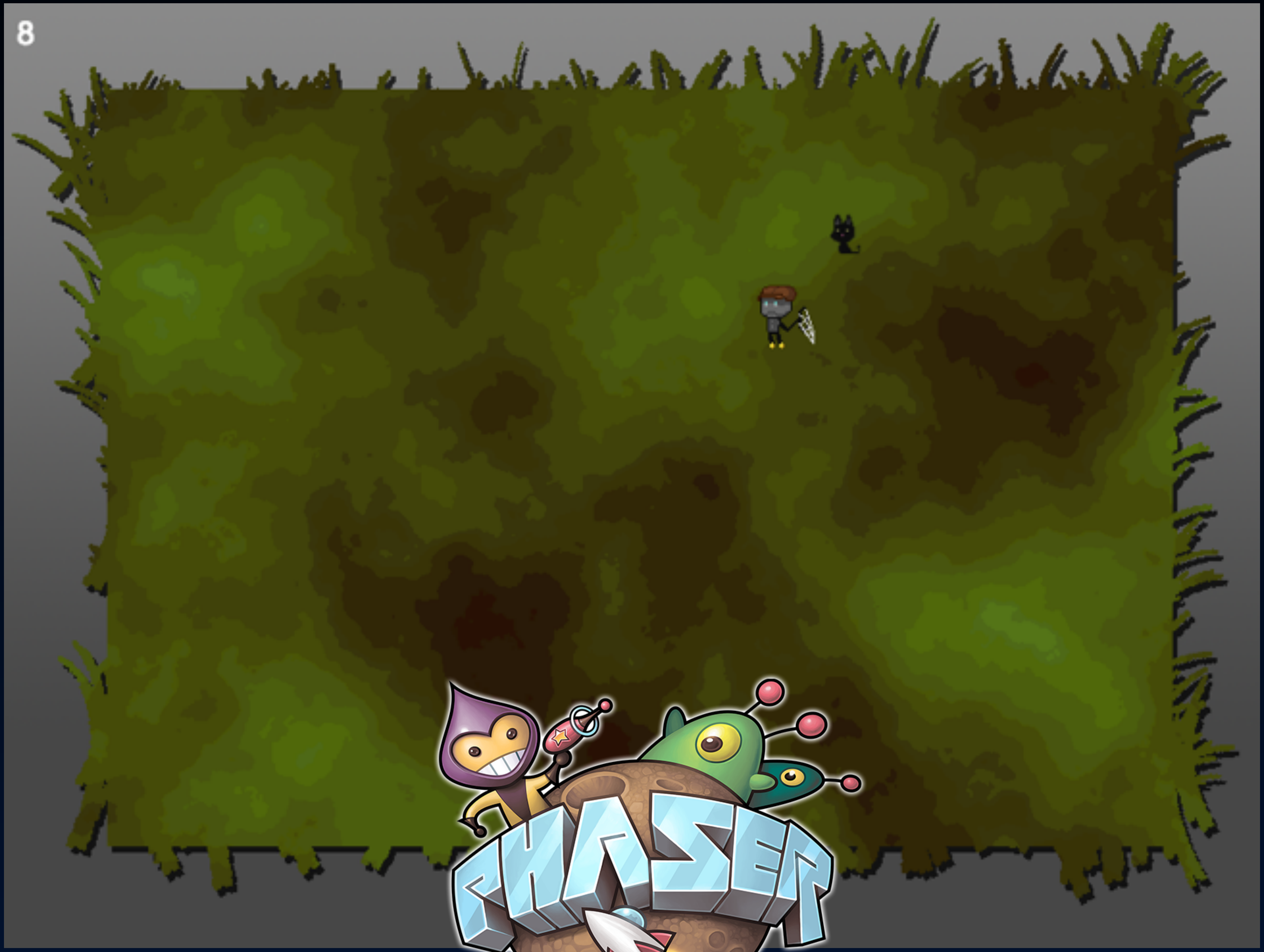
For a command-line usage, check `esvalidate` from Esprima package (for Node.js). There is also a plugin for Grunt called grunt-jsvalidate. Ant users can take a look at an exemplary Ant task for syntax validation.

esprima.org offers live JavaScript code validation.

Software for Part 1

http://phaser.io/

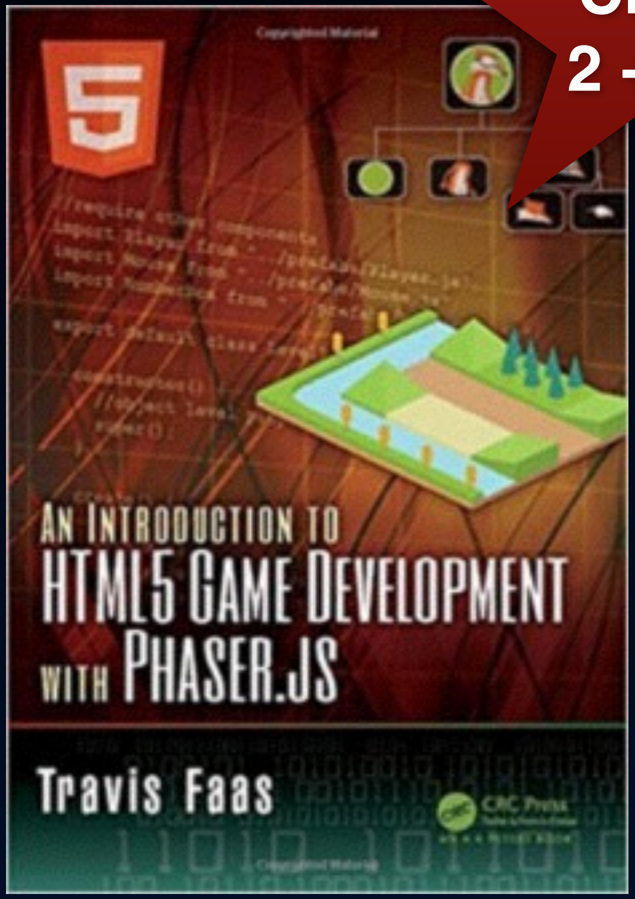
A Simple Game



Code Demo

Required Textbook for Part 1

Chs
2 + 3



An Introduction to HTML5 Game Development with Phaser.js
Travis Faas
ISBN 978-1-138-92184-9 print
ISBN 978-1-315-31921-6 ebook

Optional Textbook for Part 1

Chs 3 + 4

An Introduction to HTML5 Game Development with Phaser.js
Travis Faas
ISBN 978-1-138-92184-9 print
ISBN 978-1-315-31921-6 ebook

What to Do Next

- **Required Reading**
  - *An Introduction to HTML5 Game Development with Phaser.js*
    - Chapter 2: State of HTML5 Games
    - Chapter 3: A Simple Game
- **Optional Reading**
  - *Eloquent JavaScript* free eBook: *http://eloquentjavascript.net/*
    - Chapter 3: Functions
    - Chapter 4: Data Structures: Objects and Arrays
- **Homework**
  - **Cloud9:**
    set up your class workspace – *homework files must be posted there!*
  - **Assignment 2: First Game**
  - **Discussion 1: Check-in Discussion:**
    Make sure to respond with a post to a classmate this week.
  - Homework due **uploaded** to Canvas by **11:59pm Tuesday 9/5**
- **Canvas Site:** class videos, article links and lecture materials available
  - *https://canvas.santarosa.edu/courses/25555*