

10% - //HTML: code used inside the <body>

Choose any TEN options

- ___ Contains one or more linked images:
- ___ Contains a button: <button>
- ___ Contains a text input line: <input>
- ___ Contains a text input area: <textarea>
- ___ Contains a link to another document:
- ___ Contains a header: <h1> <h2> <h3> <h4> <h5> <h6>
- ___ Contains a paragraph: <p>
- ___ Contains a line break:

- ___ Contains strong or emphasized text:
- ___ Contains a span or div: <div>
- ___ Contains a list of items:
- ___ Contains a table: <table>
- ___ Contains an HTML canvas: <canvas>
- ___ Contains Scalable Vector Graphics: <svg>
- ___ Contains a playable mp3 file: <audio>

___ Contains other HTML as approved by instructor

10% - //CSS: code used inside <style>

Choose any TEN options

- ___ Apply a style to a tag: p { ... }
- ___ Apply a style to an ID: #exampleID { ... }
- ___ Apply a style to a class: .exampleClass { ... }
- ___ Sets the family of text: font-family: ...
- ___ Sets the color of text: color: ...
- ___ Sets the background color of text: background-color: ...
- ___ Sets the size of text: font-size: ...
- ___ Sets the weight of text: font-weight: ...
- ___ Sets the style of text: font-style: ...
- ___ Sets the width or height of an element: width: | height: ...
- ___ Sets the positioning of an element: position: ...
- ___ Positions an element using measurements: top: | left: ...
- ___ Sets opacity: opacity: ...
- ___ Sets CSS animation: @keyframes ...

___ Contains other CSS as approved by instructor

5% - //INIT: code used inside the <script>

Choose any FIVE

- ___ Uses numerical variables: exampleNumber = 42;
- ___ Initializes a numerical variable as 0: exampleNumber = 0;
- ___ Uses string variables: exampleString = 'Hello!';
- ___ Initializes a string variable as: exampleString = '';
- ___ Uses an array: exampleArr = ['Hello', ' world!'];
- ___ Initializes an array as []: exampleArr = [];
- ___ Uses an object: exampleObj = { language: 'English', greeting: 'Hi!' };
- ___ Initializes an object as {}: exampleObj = {};
- ___ Uses a two dimensional array: exampleArr = [[1, 2, 3], [4, 5, 6]];
- ___ Uses a complex data structure (eg: an array of objects):
exampleArr = [
 { langauge: 'English', greeting: 'Hello!' },
 { langauge: 'German', greeting: 'Guten Tag!' }
];
- ___ Uses three dimensional arrays
- ___ Uses pointers (an ugly hack, but it is kind of possible)
- ___ Contains other variable structures as approved by instructor

5% - //INPUT: trigger functions and read information

Choose any FIVE options

- ___ Trigger things by clicking on an HTML element:
document.getElementById(...).onclick = exampleFunction;
- ___ Trigger things (like the whole page) when they finish loading:
document.body.onload=exampleFunction()
- ___ Trigger things by changing the contents of an input line or area:
document.getElementById(...).onchange=exampleFunction;
- ___ Read information from an input line:
document.getElementById('inputID').value
- ___ Read information from the keyboard:
window.onkeydown = function (keyEvent) { ... }
- ___ Trigger a function by mousing over:
document.getElementById(...).onmouseover=exampleFunction;
- ___ Trigger a function by moving the mouse:
document.onmousemove = function (mouseEvent) { ... }
- ___ Trigger a function by pressing the mouse button:
document.onmousedown = function () { ... }
- ___ Read the height or width of the window:
window.innerWidth | window.innerHeight
- ___ Read the current position of an element:
document.getElementById('inputID').offsetTop | .offsetLeft
- ___ Turning off the click or mouse functions of an element:
document.getElementById('exampleID').onmouseover = ''
- ___ The program reads the time: new Date()
- ___ The user uploads a file or a link
- ___ Uses drop-down menus
- ___ Contains other input as approved by instructor

15% - //PROCESS: code inside the <script>

Choose any FIFTEEN options (sometimes can be used more than once)

- ___ Uses a singular if statement: if (exampleVar == 1) { ... }
- ___ Uses an if...else if...else chain:
if (exampleVar == 1) { ... }
else if (exampleVar == 2) { ... }
else { ... }
- ___ Uses a for loop: for (i in exampleArr) { ... }
- ___ Uses a nested loop
- ___ Uses string addition: 'Hello' + ' world!'
- ___ Uses a split function: exampleVar.split(' ')
- ___ Uses a join function: exampleArr.join(', ')
- ___ Uses arithmetic functions: 5 + 6 - 7 * 3 / 4
- ___ Uses rounding functions: Math.round() | .floor() | .ceil()
- ___ Generates random numbers: Math.random()
- ___ Uses .Math (powers, trig, etc.): Math.pow() Math.PI
- ___ Uses other string functions
- ___ Uses regular expressions
- ___ Uses other array functions
- ___ Uses other object functions
- ___ Uses time functions
- ___ Uses animation callbacks:
\$('#exampleID').animate(
 {top: '100px',
 duration: 1000,
 easing: 'linear',
 complete: function () { ... }
});
- ___ Uses setInterval/setTimeout
- ___ Uses recursion or function callbacks
- ___ Contains other process functions as approved by instructor

5% - //OUTPUT: code inside the <script>

Choose any FIVE options

- ___ Outputs inside an html tag:
document.getElementById('exampleID').innerHTML
- ___ Adds/removes/changes an image:
document.getElementById('exampleID').src
- ___ Changes the position of an element:
document.getElementById('exampleID').style.top | .left
- ___ Changes the style of an element:
document.getElementById('exampleID').style.backgroundColor
- ___ Uses animation to change position:
\$('#exampleID').animate(
 {top: '100px',
 duration: 1000, easing: 'linear'}
);
- ___ Uses animation to change other style properties:
\$('#exampleID').animate(
 {width: '100px',
 duration: 1000, easing: 'linear'}
);
- ___ Creates a pop up window: alert('Hi there!');
- ___ Sends data to the developer's console:
console.log('in mainProcedure...');
- ___ Plays an audio clip:
document.getElementById('audioID').play();
- ___ creates a new element:
document.createElement('img')
- ___ appends an element to an ID:
document.getElementById('outputID').appendChild(
 newElement
);
- ___ Uses other output as approved by instructor

20% - //Programming structure

Use ALL TEN options (each skill has double weight)

- Uses a function:
var exampleFunction = function () { ... }
- Use a function that accepts parameters/arguments as input:
var exampleFunction = function (exampleVariable) { ... }
- Uses a function that returns information:
return exampleVariable
- Uses an object-oriented structure with a property:
var exampleObject = { exampleProperty: 'example' }
- Uses an object-oriented structure with a method:
var exampleObject = { exampleMethod: function () { ... } }
- Uses .this inside an object-oriented structure:
this.row = this.row + 1;
- Uses an object constructor:
var ExampleConstructor = function (exampleVar) {
 this.exampleProperty = exampleVar;
};
- Uses separate functions/methods for reading input
- Uses separate functions/methods for processing
- Uses separate functions/methods for outputting information
- Optional: Uses other program structure as approved by instructor

10% - //CHECKLIST

- x 10 Completes and hands in an exam skills checklist

20% - //STYLE: Professional programming habits and communication

- ___ The author of the program is indicated in the <head>
- ___ Comment for the date started and dates modified, including what was modified
- ___ Comment for describing the whole program
- ___ Comment for describing each //INIT: block, function or method
- ___ Comment for describing each //INPUT: function or method
- ___ Comment for describing each //PROCESS: function or method
- ___ Comment for describing each //OUTPUT: function or method
- ___ Descriptive comments for each function or method
- ___ Descriptive comments for each loop
- ___ Descriptive comments for each if statement
- ___ Uses descriptive, full language variable names
- ___ Uses descriptive, full language function and method names
- ___ Uses camelCase for variable and function names
- ___ Uses UpperFirst for object constructor names
- ___ Avoids lines longer than 96 characters
- ___ Creates sub-functions/methods if a function/method is over 20 lines (ask if an exception is needed...)
- ___ Accurately indents code
- ___ groups program code by: init, main , input, process, output
- ___ Orders object code structures by: init, main , input, process, output
- ___ Separates HTML clearly into head (meta/script/style) and body

Name: _____

Advice

- Get things working in bite-sized chunks.
- Use console logs to tell you where you are in your code.
- Use console logs to tell you what values your variables contain.
- Do CSS work when you are stuck or need a break.
- Plan your data structures carefully before you start.
- Have a clear definition of what you want input and what you want to output. This makes the process much easier.
- When looking online for advice, keep in mind that the quality of publicly available code is iffy. Get me to take a quick look at it...