VARIABLES

Names, types, assignment

- Variables store data
  - The value of the data may vary (hence named variable)

- Programmer (you) creates variable
  - Declaring a variable creates it
  - Variables have a value and type (number, string, ...)

```javascript
var GPA;
GPA = 3.9;
```
Identifiers (names): rules

- Composed of any combination of
  - Letters: a, A, x, Z, ...
  - Digits: 0, 3, 23, ...
  - Underscore: _
- May be of any length
- Are case-sensitive: (A \( \neq \) a)
  - Sum, sum, SUM: all different
- Can not begin with a digit
- Can not be a reserved word
  - var, document, ...
  - About 140 reserved words (don’t need to memorize)

Valid/invalid?

- x
- 3D
- my1stProgram
- rate of pay
- R2D2
- $PerHour
- _rate_of_pay_2005
- document

Identifiers (names): guidelines

- Mnemonic
  - ✔ salary, interestRate, I
  - ✗ Fred, GoHeels, UNC74DUKE70
- Start with lower case
- Use _ or capitals for readability
  - ✔ rate_of_pay, rateOfPay
  - ✗ rateofpay
- Not too long
  - ✗ NumberOfPeanutButterScoopsThatNeedToBeAdded
Data types: numbers

- **Numbers**
  - Integers: 1 0 -37 1000
  - Real: 1.0 -37.5 1.23e8 123000000

- **Caution: computer numbers ≠ math numbers**
  - Math: 1/3 is 0.33333333... ∞
  - Computer: storage is finite
    - Computer 1/3 ≠ Math 1/3
    - Same with: Instead use:
      - dollars = 0.1;
      - cents = 10;

Data types: strings and booleans

- **String:** “...”
  - A sequence of characters
- **Escape for special characters**
  - \" : double quote
  - \\ : single quote
  - \ : backslash
  - \t : tab
  - \n : new line

- **How many characters in:**
  - “Hello”
  - “He said, \"Help!\"”
  - “”

- **Boolean:**
  - Logic values
    - true
    - false
  - For decision making
Variables

- A variable has
  - Name
  - Type (number, string, boolean)
    - In JS, type of a variable can change
    - In some languages, type is fixed (strongly typed)
  - Value (can change)
  - Location in memory (invisible to us)

- Examples:
  - GPA
    - type: number
    - value: 3.8
  - facultyName
    - type: string
    - value: Jasleen

Assigning/changing values & types

- Declaring:
  - var GPA; // created; no value or type set yet

- Initializing:
  - var name = init value;
    - var rate = 5; // type: number
    - var facName = “Jasleen”; // type: string
    - var answer = true; // type: boolean

- Assignment statement (most common):
  - var = expression;
    - 1. Expression is evaluated
    - 2. Value stored in variable; old value is lost
      - X = 20;
      - X = 20 + 30 + 40;  
      - 90 20 10
Array variables

- How to represent a set of quantities, $S_0, S_1, S_2, \ldots$?  
  - Can use an array (an ordered set of variables)

- Examples:
  ```javascript
  var S = new Array(); // an empty array
  S[0] = 1;
  S[1] = "jasleen"; // different types of elements
  var i = 2;
  S[i] = S[0] + 10; // referencing elements
  ```

- Multi-dimensional array examples:
  ```javascript
  var chessBoard = new Array(8);
  chessBoard[0] = new Array(8); // an array of arrays
  chessBoard[0] = ["R","N","B","Q","K","B","N","R"]; // assigning all at once
  chessBoard[0][3] = "Q"; // assigning individually
  ```

EXPRESSIONS

Numerical, string, boolean
Numerical expressions

- Consist of
  - **Operands**: numbers, variables with numeric value
  - **Operators**: +, -, *, /, %, ( )

- e.g.,

<table>
<thead>
<tr>
<th>expression</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1+2+3</td>
<td>6</td>
</tr>
<tr>
<td>3*4</td>
<td>12</td>
</tr>
<tr>
<td>3–10</td>
<td>-7</td>
</tr>
<tr>
<td>10/4</td>
<td>2.5</td>
</tr>
<tr>
<td>22%5</td>
<td>2     // operands must be integers</td>
</tr>
</tbody>
</table>

Evaluation order

- What is the value of: 3 + 4 * 5
  
  - 35? 23?

- Evaluation order:
  - *, / before +, -
    
    - 3 + 4 * 5 is 23
  - Then, left-to-right
    
    - 100 + 100 / 5 / 2
      
      - 20
      
      - 10
      
      - 110

- Can override with ( )
  
  - (3+4)*5 is 35
More examples

```
var x = 10;

<table>
<thead>
<tr>
<th>expression</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>x + 1</td>
<td>11</td>
</tr>
<tr>
<td>x / 2</td>
<td>5</td>
</tr>
<tr>
<td>x % 3</td>
<td>1</td>
</tr>
<tr>
<td>x * x * x</td>
<td>1000</td>
</tr>
<tr>
<td>x / x</td>
<td>1</td>
</tr>
</tbody>
</table>
```

Back to assignments

- `var = expression;`
  1. Expression is evaluated
  2. Value stored in variable; old value is lost
     
     ```
     y = x + 1;
     ```

- **Note the use of** `=`
  - Not an equation (simply an assignment)
    
    ```
    x = x + 1;
    ```

- **Output/print the value of a variable:**
  
  ```
  alert(x); // will open up an alert box
  document.write(x); // will appear on HTML page
  ```
Trace exercise (play computer)

Current value of:

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>var x = 10;</td>
<td>10</td>
<td>not declared</td>
</tr>
<tr>
<td>var y = 20;</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>var z = 30;</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>x = y + z;</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>z = 0;</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>x = x - 2;</td>
<td>48</td>
<td>20</td>
</tr>
</tbody>
</table>

String expression

- Only one operation: concatenation
  
  ```javascript
  var firstName = "Jasleen"
  var lastName = "Kaur"
  var name;
  name = firstName + lastName; // no space
  name = firstName + " " + lastName; // better
  ```

- Useful property of a string: length
  
  - Number of characters in string
    ```javascript
    var s = "Hello";
    alert(s.length); // what will this show?
    s = s + s;
    alert(s.length); // what will this show?
    s = "";
    alert(s.length); // what will this show?
    ```
Boolean expressions

```javascript
var x = true;
var y = false;
var list = new Array( false, true, true);
```

- **Operations on boolean variables:**
  - **OR:** \( x \lor y \)
    - Evaluates to true if \( x \) and/or \( y \) is true
    - Evaluates to false if both \( x \) and \( y \) are false
  - **AND:** \( x \land y \)
    - Evaluates to true if both \( x \) and \( y \) are true
    - Evaluates to false otherwise
  - **NOT:** \( \neg x \)
    - Evaluates to true if \( x \) is false
    - Evaluates to false if \( x \) is true

Expressions that yield boolean value

- \( x == y \)
- \( x > y \)
- \( x >= y \)
- \( x < y \)
- \( x <= y \)
- \( x != y \)

- **What if you want to check if:** \( 0 \leq x \leq 100 \)
  - \( 0 \leq x \leq 100 \)
  - \( 0 \leq x \leq 100 \)

- **Expressions can be arbitrarily complex**
  - \( x > y \land z == 3 \land (\text{name}==“Aaron” \lor y>5) \)
Tip: short-circuit evaluation

- What is value of:
  \[(3 > 2 \quad \text{||} \quad \ldots)\]
  \[(3 < 2 \quad \text{&&} \quad \ldots)\]

- Useful strategy:
  - Go left-to-right, until you know the answer for sure

Mixed expressions

- Javascript is forgiving
  - Java is rigid

- Examples:
  - \[3 \times \text{“40”} \quad \text{is} \quad 120\]
  - \[3 + \text{“40”} \quad \text{is} \quad \text{“340”}\]
  - How to get 43?
    - \[3 + 1 \times \text{“40”} \quad \frac{40}{\text{43}}\]
  - \[3 + \text{“four”}\]
  - \[3 \times \text{“four”} \quad \text{is} \quad \text{NaN (not defined)}\]
Changing type

- **JS is not strongly typed**
  - Allows type of a variable to be changed
    ```javascript
    var x = 5; // type: number
    x = 10;   // type: number
    x = "Jasleen"; // type: string
    ```

- **Effect on compiler:**
  - Doesn’t know how to translate: `z = x + y;`
    - Add or concatenate?
    - Will depend on values of `x` and `y`
      - Which may be known only when the program runs (next slide)

Input from outside world

- **Another way of setting value of variable:** `prompt`
  - Opens a prompt window
  - Returns a value of type string
    ```javascript
    var name = prompt("Enter your name.");
    var age = prompt("Enter your age.");
    age = age + 1;
    alert(name + ", next year you will be " + age);
    ```
  - If user enters an age of 41, what will the alert box say?
    - Let’s try it...

- **Boolean values:** `confirm`
  - Opens a confirm window: OK (true) or cancel (false)
    ```javascript
    var answer = confirm("Do you want some fries?");
    ```
PRACTICE

Let's use whatever we’ve learnt so far...

Cash register

- Single item purchase with cash
  - Prompt for purchase cost of item
  - Add 7% sales tax
  - Show total
  - Prompt for amount of cash given by customer
  - Show change due

- Will code work always?
HTML vs Javascript

- Very different!
  - HTML: markup language (for editing)
  - JS: programming language (for computing)

- JS code is included in an HTML file
  - But only within the `<script>` `</script>` tags!

- HTML text exists in the same file
  - But only outside the `<script>` `</script>` tags!