Top-down approach

- So far, seen some JS syntax
  - But how do you solve a problem using it?

- Top-down approach:
  - Divide problem into smaller problems
  - Stress WHAT over HOW
  - Keep dividing, until WHAT \( \rightarrow \) HOW is easy

- Example: payroll

  Becomes documentation (comments)
Recall building blocks

- Sequence of instructions
  - Assignment
  - I/O
    - alert
    - prompt
    - confirm
    - document.write

- Selection
- Iteration
- Sub-routine (suspend and resume)
  - Also known as (AKA): functions, procedures, methods

Recall suspend-and-resume

- Subroutines rely on the suspend-and-resume paradigm
Functions

```
function name()
{
    ... code ...
}
```

- **What?**
  - Code
  - Separate from “main” program
    - “main” – part that executes automatically
  - Functions: executed only when explicitly called

- **Example:**
  ```javascript
  function myAlert()
  {
    alert("Hello there");
  }
  ```

Location

```
<head>
<script type="text/javascript">
/*!--
    function myAlert()
    {
        alert("Hello there");
    }
    -->
</script>
</head>
```

- **Function definition:**
  - Put within `<script>` in header (within `<head>`)  
  - Code is not executed when encountered by browser

- **Function call:**
  - In `<body>` within `<script>` (referred to as “main” code)
    ```javascript
    myAlert(); // suspends to execute function code, 
    // returns & resumes after function is 
    // executed
    ```
What causes sub-routines to end?

1. Reaches end of code “}”
   - It runs out of function code to execute

2. Explicit return statement
   - return;
   - Often, in an if statement:
     - if (done) return;

3. Error

   □ Where does it return?
   - To position from where it was called