(require spd/tags)
(require 2htdp/image)
(require 2htdp/universe)

; My world program (make this more specific)
(@htdw WS)
; -----------------------------------
; Constants:

; -------------------------------
; Data definitions:
(@htdd WS)
; WS is ... (give WS a better name)

; -------------------------------
; Functions:
(@htdf main)
(@signature WS -> WS)
; start the world with (main ...)

(@template-origin htdw-main)
(define (main ws)
  (big-bang ws
    (on-tick tock) ;WS -> WS
    (to-draw render) ;WS -> Image
    (on-mouse ...) ;WS Integer Integer MouseEvent -> WS
    (on-key ...))) ;WS KeyEvent -> WS
)

(@htdf tock)
(@signature WS -> WS)
; produce the next ...
; !!!
(define (tock ws) ws)

(@htdf render)
(@signature WS -> Image)
; render ...
; !!!
(define (render ws) empty-image)
## Choosing form of data definition

<table>
<thead>
<tr>
<th>When the form of the information to be represented...</th>
<th>Use a data definition of this kind</th>
</tr>
</thead>
<tbody>
<tr>
<td>is atomic</td>
<td>simple atomic data (String, Number...)</td>
</tr>
<tr>
<td>is numbers within a certain range</td>
<td>number type and CONSTRAINT</td>
</tr>
<tr>
<td>consists of a fixed number of distinct items</td>
<td>enumeration (one-of several strings)</td>
</tr>
<tr>
<td>is comprised of 2 or more subclasses, at least one of which is not a distinct item</td>
<td>itemization (one-of several subclasses)</td>
</tr>
<tr>
<td>consists of items that naturally belong together</td>
<td>compound data</td>
</tr>
<tr>
<td>is arbitrary sized</td>
<td>well formed self-referential data definition (or mutually referential)</td>
</tr>
<tr>
<td>is naturally composed of different parts</td>
<td>reference to another defined type</td>
</tr>
</tbody>
</table>

## Data Driven Template Rules

<table>
<thead>
<tr>
<th>Form of data</th>
<th>cond question (if any)</th>
<th>Body or cond answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>atomic non-distinct</td>
<td>type predicate (string? x) (number? x) etc.</td>
<td>(... x)</td>
</tr>
<tr>
<td>atomic distinct</td>
<td>equality predicate (string=? x “red”) etc. possible w/ guard</td>
<td>(...)</td>
</tr>
<tr>
<td>one of</td>
<td>predicate (firework? x)</td>
<td>cond w/ one Q&amp;A pair per subclass be sure to guard in mixed data itemizations</td>
</tr>
<tr>
<td>compound</td>
<td></td>
<td>all selectors (... (balloon-x b) (balloon-y b))</td>
</tr>
<tr>
<td>self-reference</td>
<td></td>
<td>form natural recursion (fn-for-los (rest los))</td>
</tr>
<tr>
<td>reference</td>
<td></td>
<td>call to other type’s templates function (fn-for-drop (first lod))</td>
</tr>
</tbody>
</table>

## Metadata tags

- @assignment
- @cwl
- @problem
- @htdw
- @htdd
- @htdf
- @signature
- @dd-template-rules
- @template-origin
- @template

For additional parameters with atomic type add parameter everywhere after ...