Inform7 Cheat Sheet

v0.2

by Mark-Oliver Reiser

March 2011

most examples taken from
Writing with Inform by Graham Nelson
(part of the Inform 7 documentation)

and
Inform7 for Programmers by Ron Newcomb
**Objects and Kinds of objects:**

- **object**
  - singular-named/plural-named
  - improper-named/proper-named
    - printed name (text)
    - printed plural name (text)
    - indefinite article (text)

- **thing**
  - unlit/lit
  - inedible/edible
  - portable/fixed in place
  - described/undescribed
  - unmarked for listing/marked for listing
  - mentioned/unmentioned
  - scenery: wearable
  - pushable between rooms
    - description (text)
    - initial appearance (text)
    - matching key (object)

- **room**
  - lighted/dark
  - unvisited/visited
    - description (text)
    - map region (object)

- **region**
  - unmarked for listing/marked for listing
    - opposite (direction)

- **direction**

- **person**
  - male/female
    - carrying capacity (number)

- **man**
  - male/female
  - neuter

- **woman**
  - female/male
  - neuter

- **animal**

- **vehicle**
  - enterable
  - fixed in place
  - portable

- **player's holdall**
  - enterable
  - fixed in place

- **container**
  - opaque/transparent
  - open/closed
  - unopenable/openable
  - unlocked/locked
  - enterable
  - lockable
  - carrying capacity (number)

- **supporter**
  - fixed in place
    - portable
  - switched off/switched on

- **device**
  - fixed in place
  - portable
  - unopenable/unopenable
  - unlocked/locked
  - lockable
  - pushable between rooms
  - other side (object)

- **door**
  - fixed in place
    - portable
    - unopenable/unopenable
    - unlocked/locked
  - lockable
  - pushable between rooms

- **backdrop**
  - fixed in place/portable
  - scenery
  - pushable between rooms

**Building blocks of Inform7 source text:**

**Phrases** are the fundamental building blocks.

Some phrases may stand alone in the source (esp. those stating facts about the game world).

The entrance hall is a room.

Other phrases may not appear alone, because Inform7 does not know when to execute them. **Rules** are one way to specify when to execute a phrase.

Instead of **taking a large container**:

say "[The noun] seems a bit too heavy."

**Phrase** (in a rule)

**Phrase** (stand alone)

**Rule**

**Action**

**Description**

**Text replacement**
Phrases
Phrases are the fundamental building blocks of Inform7 source text; they can have very different uses.
A phrase ends with a full stop which may be omitted if the phrase ends with a string ending with ":" or ":s". Several phrases can be concatenated into a composite phrase by separating them with a semicolon.

Let X be 5.
Say "Bob".
Say "Hi." 
Say "Hello"; say "world."

Phrases to state facts about the game world
The Lobby is a room.
The cave is a room.
The treasure chest is a locked container in the cave.

Phrases to do something
Say ...
Move ...
Now ...
Change ...

Phrases to take a decision
If <condition> then <phrase>;
If <condition> then <phrase>; otherwise <phrase>;
If <condition>, <phrase>; otherwise <phrase>;
If <condition>, <phrase>; otherwise <phrase>;
If <condition> begin; [may all appear on a single line]
otherwise if <condition>;
<phrase>;
otherwise;
<phrase>;
end if.

If <condition>;
otherwise if <condition>:
<phrase>;
otherwise:
<phrase>;

If <variable> is:
  <value>: <phrase>;
  <value>: <phrase>;
  otherwise: <phrase>.

Phrases to repeat other phrases
Repeat with <variable> running from <value> to <value> begin;
<phrase>;
end repeat.
Repeat with <variable> running through <description of values> begin;
<phrase>;
end repeat.
Repeat through <table> begin;
<phrase>;
end repeat.
While <condition> begin;
<phrase>;
end while.

Phrases to compute a value
(called to "decide" a value in Inform7 parlance)
square root of 16

Phrases to introduce local variables
Let <variable> be <value>.
Let X be 5.

Custom phrases
Use {variables} in phrase pattern to introduce parameters:
To plainly greet (friend - a person): ...
To simply/plainly greet (friend - a person): ...
To simply/plainly greet (friend - a person): ...
Say "Hi [friend]."

Custom conditions:
To decide whether <phrase pattern>: <phrases>; decide yes/no. OR decide on <truth value>.
To decide whether danger lurks: if in darkness, decide yes; no.
To decide whether (pants - a thing) is/are on fire: decide on whether or not a random chance of 1 in 2 succeeds.

Custom value computations:
To decide which/what <returned kind> is <phrase pattern>: <phrases>; decide on <value>.

Custom say phrases:
To say <phrase pattern> <phrases>.
To say He/She for (P - a person):
if P is female, say "She"; otherwise say "He".
say "[He/She for Chris] glances at you.";

Rules
A rule consists of a preamble and one or more phrases. The preamble tells Inform when to execute the phrases.
If the preamble begins with Before, After, Instead of, Every turn, or When, and there is only one phrase, an alternative notation is available:

Creating a rule
(1) creating an unnamed rule put in a rulebook:
A/an <rulebook name> rule: <phrases>. 

(2) creating a named rule put in any rulebook:
This is the <rule name> rule: <phrases>. 

(3) creating a named rule put in a rulebook:
A/an <rulebook name> rule (this is the <rule name> rule): <phrases>.

Circumstances when putting a rule in a rulebook
first/last <rulebook> about/for/on/with rule <phrase description> while/when <condition> during <scene> : <phrases>.
If an every turn rule (this is the greet rule): say "Hi." Every turn (this is the greet rule): say "Hi.

Instead of kissing Clark, say "No thanks." Instead rule about kissing Clark: say "No thanks."
### Built-in rulebooks

<table>
<thead>
<tr>
<th>when play begins</th>
<th>when play ends</th>
</tr>
</thead>
<tbody>
<tr>
<td>every turn</td>
<td></td>
</tr>
</tbody>
</table>

- **before**
- **instead**
- **after**

- **viability**
  - **does the player mean persuasion**
  - **unsuccessful attempt**
  - **reaching inside**
  - **reaching outside**

- **for each action:**
  - **check <action name>**
  - **carry out <action name>**
  - **report <action name>**

- **for each scene:**
  - when <scene name> begins
  - when <scene name> ends

- **for each activity:**
  - before <activity name>
  - for <activity name>
  - after <activity name>

### Ending a rule

Ending a rule with an outcome – no matter if "success" or "failure" – will immediately end the entire rulebook, i.e. no more rules will be processed.

- **rule succeeds.**
  - [end rule with outcome success]
  - [end rule with outcome fail]
  - [end rule with outcome failure]

- **rule fails.**
  - [end rule with outcome fail]
  - [end rule with outcome failure]

- **make no decision.**
  - [end rule with no outcome]

### Rulebooks can define other, so-called named outcomes, that look like phrases. E.g., the visibility rulebook defines:

- **there is insufficient light:**
  - [end rule with outcome failure]

- **there is insufficient light:**
  - [end rule with outcome failure]

### Phrases for rules

- **follow <rule>**
  - [apply rule]
  - [throw away result]

- **follow <rulebook>**
  - [for <value>]
  - [apply rule based rulebook producing <value>]

- **consider <rule>**
  - [for <value>]
  - [apply rule]

- **abide by <rule>**
  - [for <value>]
  - [apply rule]

- **anonymously abide by <rule>**
  - [for <value>]
  - [apply rule]

- **rule succeeds with result <value>**
  - [end rule with outcome success]

### Custom rulebooks

- **<rulebook name> is a <value> based rulebook producing <value>**
- **The <rulebook name> has default failure success no outcome.**
- **The <rulebook name> has outcomes <outcomes>**

### Adding rulebook variables (global to all rules in the rulebook):

- **<rulebook name> has a <kind>/ <type> called <variable name>**

### Appraisal rulebook is a rulebook.

- The cat behavior rules are a rulebook producing an animal.

### The probability rules have outcomes it is likely, it is possible and it is unlikely.

### Descriptions

A description identifies one or more objects. It can be used when creating new objects (a new object will be created such that it matches the description) and when denoting existing objects.

- **<adjective> <noun>**

### Nouns

The **noun** is the name of a kind, the name of an object or one of eight special nouns (with only three different meanings):

- **something = anything**
  - equivalent to the noun thing
- **someone = somebody**
  - equivalent to the noun person
- **somewhere = anywhere**
  - equivalent to the noun room

### the open wine cask

- **something**
  - **open + noun wine cask**
  - [the noun thing + the adjective portable]

### Built-in Adjectives

#### Visible

<table>
<thead>
<tr>
<th>visible</th>
<th>touchable</th>
<th>adjacent</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>each</td>
<td>every</td>
</tr>
<tr>
<td>some</td>
<td>most</td>
<td>almost</td>
</tr>
<tr>
<td>none</td>
<td>no</td>
<td>nobody</td>
</tr>
</tbody>
</table>

- a light cloak [which is] worn by a woman [who is] in a dark room

### Custom adjectives

1. ordinary boolean or many-valued properties on kinds
   - A person can be grumpy or happy.

2. definitions:
   - **Definition:** A <kind> is <adjective> rather than <adjective>
     - if <condition>.
   - **Definition:** A room is occupied if a person is in it.
     - **Definition:** A container is large rather than small if its carrying capacity is 20 or more.

3. Comparatives/superlatives are automatically generated:
   - the largest container

### Actions

**An action** is something the player's character can do to interact with the game world, e.g. moving around, taking things, opening a door.

- **<verb in present participle> <noun>**
  - **in <room>**
  - **in the presence of <person>**
  - **while when <condition> during <scene>**
  - **for the <th> time for <n> turns**

- **doing something other than <action>**
  - [all actions except <actions>]
  - taking the ivory key
  - taking something
  - eating in the Gardens
  - eating something in the presence of Lady Bracknell

- **examining the key for the second time**
  - [spelling n out is ok up to 12]

- **examining the key for the 20th time**
  - [only if consecutive]

- **waiting for four to six turns**
  - waiting for more than 13 turns

### Going

**The action "going" happens when the player's character moves between rooms. Some special forms simplify common cases:**

- **going from <room>**
  - [when leaving the room]
  - [when entering the room]
  - [when entering from 2nd room]

- **going to <room>**
  - [when leaving the room]
  - [when entering the room]

- **going from <room> to <room>**
  - only if <rule exists>
  - [also if route does not exist!!]

- **going through <door>**
  - [player tries unmapped room]

- **going <direction> from <room>**
  - [player tries unmapper]

- **going nowhere from <room>**
  - (note the "from" here, which is inconsistent with previous lines)
Kinds of actions

Understanding

Phrases related to actions

Custom Actions

Understanding

Relations

Built-in relations

Custom relations

How to use relations

Where to use relations

(1) in assertions
(2) in conditions
Phrases for relations

next step via <relation> from <object> to <object>
number of steps via <relation> from <object> to <object>

These are special to value relations (i.e. non-object on one or both sides):

<kind> to which/whom <value> relates by <value relation>
<kind> that/which/whom <value> relates to by <value relation>

list of <kind> to which/whom <value> relates by <value relation>
list of <kind> that/which/whom <value> relates to by <value relation>
list of <kind> that/which/who relates to <value> by <value relation>
...

(repeat through these special relations)

Main Cycle

A game of interactive fiction can be seen as an endless cycle of processing actions (either triggered by player-typed commands or by "try" phrases in the source) resulting in messages being printed out and/or changes being effected to the game world.

Tables

Creating a table / list

Table <number> / Table of <name> / Table <number> = <name>
<name column #1> TAB <name column #2> ...
<value> / / <kind> TAB <value> / / <kind> ...
...

with <number> blank rows

TAB = one or more tab chars ; <kind> only in first row of empty column

Table 2.1 - Selected Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Symbol</th>
<th>Atomic number</th>
<th>Atomic weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Hydrogen&quot;</td>
<td>H</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>&quot;Iron&quot;</td>
<td>Fe</td>
<td>26</td>
<td>56</td>
</tr>
<tr>
<td>&quot;Zinc&quot;</td>
<td>Zn</td>
<td>30</td>
<td>65</td>
</tr>
<tr>
<td>&quot;Uranium&quot;</td>
<td>U</td>
<td>92</td>
<td>238</td>
</tr>
</tbody>
</table>

Referring to entries of a table

Referring to a table (written as <table> below):
the Table <number> / of <name>
the Table 2.1
the Table of Selected Elements

Referring to entries in a table (written as <table entry> below):

<column name> in row <number> of <table>
<column name> of <table>
<column name> corresponding to <column name> of <value> in <table>

Choose the row <number> in/from <table>
Choose the row with <column name> of <value> in/from <table>
Choose the blank row in/from <table>
Choose the random row in/from <table>

After choosing a row, we can refer to it simply with:
<column name> entry

Reading from & writing to a table

As usual, you can use these <table entry> references to read a value from the table or write a value to a table:

<table entry>
<table entry is <value>.
<table entry is "value".
[symbol in row 3 of Table 2.1] replaces "Zn" with "Cn"
[symbol in row 3 of Table 2.1] is "Cn" [retrieves symbol of "Fe" in Table 2.1 [retrieves "Iron"]

number of blank/filled rows in/from Table <number> of <name>
repeat through <table> : ...
repeat through <table> in reverse order : ...
repeat through <table> in reverse <column name> order : ...

blank out <table entry>
blank out the whole row / the whole <column name> column in Table <number> of <name>
blank out the whole of Table <number> of <name>

Defining objects and values with tables

<description> are defined by <table>,
<kind of values> are defined by <table>
See Writing with Inform, Chapter 13.13 for details.

Miscellaneous

Names

Names may contain spaces:
The launching base is a room.

In names of rules, rulebooks, actions etc. the keyword rule, rulebook, action, resp. is optional if not required to avoid ambiguities:
the advance time rule = advance time
the instead rules = the instead rulebook = instead

Conditions

<table>
<thead>
<tr>
<th>rule succeeded/failed</th>
<th>true if most recently followed rule ended in success / failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>outcome of the rulebook</td>
<td>true if the outcome of the most recently followed rulebook is &lt;named outcome&gt;</td>
</tr>
<tr>
<td>&lt;value&gt; relates to &lt;kind&gt; by &lt;value relation&gt;</td>
<td>retrieves value &quot;Zn&quot;</td>
</tr>
<tr>
<td>&lt;value&gt; relates to &lt;value&gt; by &lt;value relation&gt;</td>
<td>retrieves value &quot;Zn&quot;</td>
</tr>
<tr>
<td>there is a &lt;table entry&gt;</td>
<td>there is no &lt;table entry&gt;</td>
</tr>
<tr>
<td>&lt;table name&gt; is empty</td>
<td></td>
</tr>
<tr>
<td>the &lt;object&gt; is a &lt;column name&gt; listed in Table &lt;number&gt; of &lt;name&gt;</td>
<td></td>
</tr>
</tbody>
</table>

What else ... ?

This Cheat Sheet intends to illustrate the basic structures in the Inform7 language, not to provide a comprehensive list of all available phrases, rules, etc. However, the Inform7 index provides such lists, esp.: Phrasebook Index Rules Index Actions Index