

## Getting Started with Inform 7

### **PART TWO: Decorating Your World**

---

The thing, scenery, backdrop, container, supporter, and device objects, their respective properties, and descriptions of those objects.

#### **The THING Object**

---

Any object defined within the world is a thing and will be portable by default. Normally, we do not have to declare objects specifically as things other than in several exceptions.

The following examples are equivalent for creating a pencil:

```
The pencil is in the Foyer.
```

```
The pencil is a thing in the Foyer.
```

```
The pencil is a thing. It is in the Foyer.
```

```
A thing called a pencil is in the Foyer.
```

This creates an object called a pencil that is located in a room called the Foyer. It can be picked up, carried, or dropped by the player but not much else.

Things, by default, will also be mentioned separately when the player enters the room where they are located:

**Foyer**

```
The Foyer is an austere space containing no furnishings.
You can see an exit to the north.
```

```
You can see a pencil here.
```

```
>
```

By default, when the player "examines" the pencil they will receive the following response:

```
>examine pencil
You see nothing special about the pencil.
```

We can create something more useful than that.

To create a more useful description of the pencil, we append a new sentence beginning with 'The description is' followed by the description nested within double quotations.

The pencil is in the Foyer. **The description is** "Freshly sharpened and covered with bite marks, it's a good old #2 yellow pencil."

This will produce the following during play:

**Foyer**

The Foyer is an austere space containing no furnishings.  
You can see an exit to the north.

You can see a pencil here.

>examine pencil

Freshly sharpened and covered with bite marks, it's a good old #2 yellow pencil.

>take pencil

Taken.

>drop pencil

Dropped.

Some standard properties for things are: **edible** (can be eaten), **wearable** (the player can wear it), **fixed in place** (it is not portable), and **lit** (provides light).

If we want to create something the player could wear:

```
The baseball cap is in the Foyer. It is wearable.
```

It will look (and behave) like this in play:

```
You can see a baseball cap here.  
  
>take cap  
Taken.  
  
>wear cap  
You put on the baseball cap.  
  
>inventory  
You are carrying:  
  a baseball cap (being worn)  
  
>take off cap  
You take off the baseball cap.
```

If we want to create something the player could eat:

```
The apple is in the Foyer. It is edible.
```

It will look (and behave) like this in play:

```
You can see an apple here.  
  
>take apple  
Taken.  
  
>eat apple  
You eat the apple. Not bad.
```

If we want to create something that cannot be moved:

The coat rack is in the Foyer. **It is fixed in place.**

It will look (and behave) like this in play:

You can see a coat rack here.

```
>take coat rack
That's fixed in place.
```

If we want to create something that provides light:

The flashlight is in the Foyer. **It is lit.**

It will look (and behave) like this in play:

**Foyer**

There is an exit to the east.

You can see a flashlight here.

```
>go east
```

**Darkness**

It is pitch dark, and you can't see a thing.

```
>go west
```

**Foyer**

```
>take flashlight
```

Taken.

```
>go east
```

```
>Kitchen
```

The Kitchen is of a modern design and remarkably clean.

There is an exit west.

Any thing that has the 'lit' property will illuminate a dark room. This is discussed further in Making Your World Behave.

## OTHER KINDS OF THINGS:

### The Scenery Object

---

We make things scenery when they are explicitly mentioned in the room description and require some description if examined.

Scenery objects are always fixed in place and will NOT be mentioned when the player enters its location.

The Foyer is a room. "This room is an austere place with only a large painting on the wall."

The large painting is scenery in the Foyer. "The painting is huge: over ten feet tall and six feet wide. It depicts a hunting party that seems to have gone terribly wrong."

Here is how it will look in play:

```
Foyer
This room is an austere place with only a large painting
on the wall.

>x painting
The painting is huge: over ten feet tall and six feet
wide. It depicts a hunting party that seems to have gone
terribly wrong.

>take painting
That's hardly portable.
```

## **The Backdrop Object**

---

This is a thing that is similar to the scenery object but can be in more than one location at a time. Things such as mountains in the distance, the sky, wind, smells, sounds, etc., would be good candidates for backdrops.

Backdrops need to be mentioned in the room description (or elsewhere) as they will not be mentioned otherwise.

The Purple Moon is a backdrop. **It is everywhere.** "The purple moon seems to follow you everywhere."

The fog is a backdrop. **It is in the driveway, the backyard, and the meadow.** "The denseness of this fog makes walking treacherous."

## The Container Object

---

Containers are things that other things can be put into and taken out of.

Containers are, by default, always open and will be mentioned separately when the player enters its location. Note: we do not need to provide descriptions for containers.

The piggy bank is a container in the Foyer.

Here is how it will look (and behave) in play:

### **Foyer**

This room is an austere place.

You can see a piggy bank (empty) and a coin here.

>x piggy bank

The piggy bank is empty.

>put coin in piggy bank

You need to be holding the coin before you can put it into something else.

>take coin

Taken.

>put coin in piggy bank

You put the coin into the piggy bank.

>x piggy bank

In the piggy bank is a coin.

>take piggy bank

Taken.

>take coin

Taken.

>x piggy bank

The piggy bank is empty.

>inventory

You are carrying:

  a coin

  a piggy bank

Containers have more sophisticated behaviors as the preceding examples illustrates. These can be further complicated with the addition of container properties.

Properties for a container object are: **open** (is open - the default but cannot be closed), **closed** (is closed but cannot be opened), **openable** (can be opened and closed by the player), **locked** (is locked and closed but can be locked and unlocked as well as opened and closed), **lockable** (is unlocked but can be locked and unlocked as well as opened and closed), and **transparent** (the player can see contents inside).

Here is a way to create a cookie jar (that contains a chocolate chip cookie) that starts off being closed but can be opened and/or closed by the player:

```
The jar is a closed openable container.
```

```
A chocolate chip cookie is in the jar. It is edible.
```

Here is how it will look (and behave) in play:

#### **Foyer**

```
This room is an austere place.
```

```
You can see a jar (closed) here.
```

```
>x jar
```

```
You can't see inside, since the jar is closed.
```

```
>open jar
```

```
You open the jar, revealing a chocolate chip cookie.
```

```
>take cookie
```

```
Taken.
```

```
>close jar
```

```
You close the jar.
```

Let's add another property: transparent.

```
The glass jar is a closed openable transparent container.
```

```
A chocolate chip cookie is in the jar. It is edible.
```

Now the player will be able to see what is inside the jar even if it's closed.

Here is how it will look (and behave) in play:

**Foyer**

This room is an austere place.

You can see a glass jar (closed) (in which is a chocolate chip cookie) here.

>x jar

In the glass jar is a chocolate chip cookie.

Let's take it a step further and lock the jar.

The glass jar is a **locked** transparent container.

OK, but how can the glass jar be unlocked? Simple. We need to create some object and state that it will unlock the jar:

The glass jar is a locked transparent container.

**The silver key is carried by the player. It unlocks the glass jar.**

Here is how it will look (and behave) in play:

**Foyer**

This room is an austere place.

You can see a glass jar (closed) (in which is a chocolate chip cookie) here.

>open jar

It appears to be locked.

>unlock jar with silver key  
(first taking the silver key)  
You unlock the glass jar.

>open jar

You open the glass jar.

>take cookie

Taken.

>close jar

You close the glass jar.

>lock jar

What do you want to lock the glass jar with?

>silver key

You lock the glass jar.

The container object can be a very a complex yet versatile part of your IF work.

## The Supporter Object

---

A supporter is any object that has the capability of having other objects placed on its surface.

Supporters are, by default, fixed in place and will be mentioned separately when the player enters its location.

```
The coffee table is a supporter in the Foyer. The
description is "This little table has been in your family
for generations and you've always hated it."
```

```
The cup is carried by the player.
```

Here is how it will look (and behave) in play:

### **Foyer**

```
This room is an austere place.
```

```
You can see a coffee table here.
```

```
>x table
```

```
This little table has been in your family for generations
and you've always hated it.
```

```
>put cup on table
```

```
You put the cup on the coffee table.
```

```
>look
```

### **Foyer**

```
This room is an austere place.
```

```
You can see a coffee table (on which is a cup) here.
```

```
>x table
```

```
This little table has been in your family for generations
and you've always hated it.
```

```
>take table
```

```
That's fixed in place.
```

Properties for supporters are: **portable** (the supporter can be carried by the player) and **enterable** (the supporter can be sat on or stood on).

A supporter, by using the 'enterable' property, can create things like chairs, sofas, beds, ladders, etc.

The recliner is an **enterable** supporter in the Foyer. The description is "The big recliner looks like it would be extremely comfortable to sit in."

Here is how an enterable supporter looks (and behaves) in play:

**Foyer**

This room is an austere place.

You can see a recliner here.

>x recliner

The big recliner looks like it would be extremely comfortable to sit in.

>take recliner

That's fixed in place.

>sit in recliner

You get onto the recliner.

>look

**Foyer** (on the recliner)

This room is an austere place.

>get up

You get off the recliner.

**Foyer**

This room is an austere place.

You can see a recliner here.

That is how we make basic beds and chairs.

One more, let's make a step-stool that we can carry wherever we might need it.

The step-stool is a portable enterable supporter in the Foyer. The description is "This little step-stool comes in handy when you need to get something that you cannot reach."

Here is how it will look (and behave) in play:

**Foyer**

This room is an austere place. The Kitchen is north of here.

You can see a step-stool here.

>x step-stool

This little step-stool comes in handy when you need to get something that you cannot reach.

>take step-stool

Taken.

>go north

**Kitchen**

The kitchen is of a modern design and remarkably clean. The foyer is to the south.

>stand on step-stool

You can only get into something free-standing.

>put down step-stool

Dropped.

>stand on step-stool

You get onto the step-stool.

>look

**Kitchen** (on the step-stool)

The kitchen is of a modern design and remarkably clean. The foyer is to the south.

There you have it - the basics of the supporter object.

## The Device Object

---

Any object that can be switched on or off.

Devices are, by default, portable, switched off, and will be mentioned when the player enters its location.

The coffee pot is a device in the foyer.

Here is how it will look (and behave) in play:

### **Foyer**

The foyer is an austere place.

You can see a coffee pot here.

>x coffee pot

The coffee pot is currently switched off.

>turn on coffee pot

You switch the coffee pot on.

>x coffee pot

The coffee pot is currently switched on.

Standard properties are **switched on** and **switched off** (the default).

The coffee pot is a device in the Foyer. It is switched on.

Devices are further discussed further in Making Your World Behave.

## CREATING NEW THINGS

Often, we need to make many objects that have the same properties. For example, we may be creating a work that will have ten different chairs and six different tables. We can create new things by combining different kinds and/or properties.

The following creates a new kind called a table (place this near the beginning of your code):

```
A table is a kind of supporter.
```

Then, when we need to create tables for our world, we can do the following:

```
The kitchen table is a table in the Kitchen.  
The dining table is a table in the Dining Room.  
The night stand is a table in the Bedroom.
```

It may seem redundant but it is often easier to remember what a table is than what a supporter is.

The following creates a new kind that we put things onto and sit on:

```
A chair is a kind of supporter that is enterable.
```

Now we can use this new chair kind to create anything that we might normally sit on:

```
The couch is a chair in the Living Room.  
The stool is a chair in the Kitchen.  
The park bench is a chair in the Meadow.
```

Again, this technique can save you many key strokes and is easier to remember what a chair is than what an enterable supporter is.

## **OBJECTS THAT ARE PART OF OTHER OBJECTS**

Let's say we want to create a night stand that has a drawer and that drawer will contain an important key for the player to find.

First let's create the night stand itself:

```
The night stand is a supporter in the Bedroom. The
description is "The night stand appears alone and small
against the wall. It has one drawer."
```

OK, but what about the drawer? We want something that can contain things and be opened (and closed). Our best bet is to make the drawer a container and then make it part of the night stand itself.

```
The drawer is a container and part of the night stand.
```

Note that we created a drawer as a container (it will contain things) and simply stated it to be part of the night stand. This makes the drawer a part of the night stand.

This will work but we need to make it a little more realistic. We'll do this by adding two properties to the drawer object:

```
The drawer is a closed openable container and part of the
night stand.
```

Openable means that the container can be opened (and will reveal anything inside) or closed (and will hide anything that is inside) by the player. Closed means that the drawer will be closed when the player first encounters it.

One last part:

```
The drawer is a closed openable container and part of the
night stand. A green key is in the drawer.
```

Now the player will find a night stand that has a drawer that they can open and find a green key inside (which can be taken).