

# Getting Started with Inform 7

## PART 2.1

### Decorating Your World: Containers

---

The container object and its properties.

A container is a kind of thing that other portable things can be put into or taken out of. A wallet, a glass vase, a bowl, a treasure chest, a desk drawer, a backpack, or a paper bag are a few examples of containers that you might want to implement.

Containers have several properties that alter their behavior as well as how the player needs to interact with them. They can get quite complex and this chapter will introduce you to the basics of the containers and how to build more complex uses for them.

Let's create a room that has a container object called a paper bag and another thing called a cupcake.

```
The Foyer is a room. "The Foyer is an austere space
containing no furnishings."
```

```
A paper bag is a container in the Foyer.
```

```
A cupcake is in the Foyer.
```

Here is how this will look in play:

**Foyer**

```
The Foyer is an austere space containing no furnishings.
```

```
You can see a paper bag (empty) and a cupcake here.
```

```
>
```

Containers are, by default, always open (allowing the player to see what is inside the container) and will be mentioned separately from the room description.

**Note:** we do not need to provide descriptions for containers.

Let's play with the paper bag and the cupcake:

**Foyer**

The Foyer is an austere space containing no furnishings.

You can see a paper bag (empty) and a cupcake here.

>examine bag

The paper bag is empty.

>take bag

Taken.

>inventory

You are carrying:  
a paper bag

>take cupcake

Taken.

>put cupcake in paper bag

You put the cupcake into the paper bag.

>examine bag

In the paper bag is a cupcake.

>drop bag

Dropped.

>look

**Foyer**

The Foyer is an austere space containing no furnishings.

You can see a paper bag (in which is a cupcake) here.

>

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

## PROPERTIES OF CONTAINERS

---

Properties for a container object are:

- **open** (is open - but cannot be closed)
- **closed** (is closed but cannot be opened)
- **openable** (is initially open but 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 initially open but can be locked and unlocked as well as opened and closed)
- **transparent** (the player can always see the contents inside the container whether it is open or closed)

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

```
The jar is a closed openable container in the Foyer.
```

```
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.

>open jar

You open the glass jar.

>take cookie

You take the chocolate chip cookie.

>

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

The glass jar is a **locked** container in the Foyer.

OK, but how can the glass jar be unlocked? Simple. We create a thing called a silver key (that will be initially carried by the player) and state that it can unlock the glass jar:

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) here.

```
>open jar  
It appears to be locked.
```

```
>inventory  
You are carrying:  
  a silver key
```

```
>unlock jar with 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 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 CARRYING CAPACITY

You may find it necessary to limit the number of things that a container can hold. We can do this by adding the carrying capacity property and assign it a value. For example:

```
The paper bag is a container in the foyer. The carrying
capacity is 1.
```

Here we have stated that the paper bag can only hold one (1) thing. Any attempts to put more than one thing into the container will result in this message:

```
>put bread in the bag
There is no more room in the paper bag.
```

To change the carrying capacity of any container, simply change the value of the carrying capacity property to any numerical value (2, 3, 4, etc.).

## PREVENTING A CONTAINER'S CONTENTS OR STATUS FROM BEING DISPLAYED

Many wish to prevent the default display of a container's status or contents from being displayed in parenthesis as in this example:

```
Foyer
This room is an austere place.

You can see a cardboard box (in which is a pencil) here.
```

To prevent containers from displaying their status or their contents in parentheses, include these two lines anywhere in your code:

```
After printing the name of a container:
omit contents is listing.
```

These two lines will prevent the status or contents of ANY container from being displayed in parentheses so that the player will have to explicitly examine the container to discover what (if anything) may be inside.

```
Foyer
This room is an austere place.

You can see a cardboard box here.

>examine cardboard box
In the cardboard box is a pencil.
```

**“BULK”**

Lastly, you probably want to avoid things like this:

**Driveway**

You can see a 2006 Toyota Camry and a paper bag (empty) here.

```
>put toyota in paper bag  
(first taking the 2006 Toyota Camry)
```

You put the 2006 Toyota Camry into the paper bag.

```
>x paper bag  
In the paper bag is a 2006 Toyota Camry.
```

```
>
```

Unless yours is a world where anything is possible, you might prefer a more realistic form of behavior. However, this is beyond the scope of these tutorials.

I would recommend downloading and installing the extension called “Bulk Limiter by Eric Eve.” You can find this extension at <http://inform7.com/extensions/Eric%20Eve/Bulk%20Limiter/index.html>