Welcome to Pluto!

Pluto is a programming environment for *Julia*, designed to be **interactive** and **helpful**.

In this introduction, we will go through the basics of using Pluto. To make it interesting, this notebook does something special: it **changes while you work on it**. Computer magic 🪄

Cats

Let's say you're like my grandma, and you have a lot of cats. Our story will be about them.

```
cat = "Ks∗;lj"
```
- `cat = "Ks∗;lj"`

Oh no! Someone messed with my pretty introduction. Change the code above to give our cat a proper name!

To edit any code, just click on it. When you're done programming, press the ⊙ in the lower-right corner of a cell to run the code. You can also use `Shift+Enter` if you are in a hurry.

UndefVarError: friend not defined

1. **top-level scope** @ *Local: 1*

- `md"I feel like our cat needs a friend. Let's call them $(friend)."`

Uh oh, what is this? I forgot to add a cell defining our friend. Can you do it for me?

A cell is a container for code & output. To add one, click on the **+** above or below another cell. You can do it wherever you like. After you're done writing code in your cell, remember to run it!

Feeding neighbors

Our cats have some neighbors. Let's involve them in the story too!

```
neighbors =
```
UndefVarError: friend not defined

1. **top-level scope** @ *Local: 1*

- neighbors = [cat, friend, "Smerfetka", "Latte"]

Now, if you're like my grandma, you're feeding the entire neigborhood by yourself. Let's see how many cans of cat food you need to prepare.

```
consumption =
```
UndefVarError: neighbors not defined

1. **top-level scope** @ ❚ *Local: 1*

- consumption = confusing_function("🥫", neighbors)

But what does `confusing_function` do? If you ever need help, click on 📚 **Live docs** in the lower right, and then place your cursor on the code you need help with.

If you don't see it, then your screen is too small! Maybe you need to zoom out?

UndefVarError: consumption not defined

1. **top-level scope** @ ❚ *Local: 1*

```
cans_in_stock = "🥫🥫🥫🥫"
```
- cans_in_stock = "🥫🥫🥫🥫"

Actually, I have a hunch there will be another cat coming. Uncomment the code below (remove the #) to add one more can. Remember to run it after making the change!

- *# cans_in_stock = "🥫🥫🥫🥫🥫"*

UndefVarError: consumption not defined

1. **top-level scope** @ ❚ *Local: 1*

Saving cats and notebooks

Alright, we have a neighborhood full of well-fed cats. But oh no, here comes...

```
scary_dog = "Piesio"
```
- scary_dog = "Piesio"

Piesio is terrorizing the neighborhood! We must do something about it!

To delete a cell like the one defining Piesio, click on the ⊗ on the right of its code.

Speaking of saving, this notebook is autosaved whenever you change something. The default location for new notebooks is D:\.julia\pluto_notebooks, you can find it using your file explorer. To change the

name or the directory of a notebook, scroll to the top - you enter the notebook's path next to the Pluto logo.

⚡ Pluto power ⚡

Oof, that dog situation was intense. Let's give our cats superpowers to make sure it never happens again!

Remember learning HTML in junior high? Now you can use it for something! Pluto lets you `@bind` variables to HTML elements. As always, every time you change something, Pluto knows what to update!

```
@bind power_level html"<input type='range'>"
```

The power level is 50, but we should do more than just say it - let's equip our cats with 50 emoji!

Multiple expressions in one cell.
How would you like to fix it?

- **<u>Split this cell into 2 cells</u>**, or
- **<u>Wrap all code in a *begin ... end* block.</u>**

1. **top-level scope** @ *none:1*

```
power_emoji = "⚡"
power = repeat(power_emoji, power_level)
```

Uh oh! Pluto doesn't support multiple expressions per cell. This is a conscious choice - this restriction helps you write less buggy code once you get used to it. To fix the code, you can either split the above cell, or wrap it in a `begin ... end` block. Both work.

We're almost done! It's time to share your amazing story. Scroll to the top of the notebook, and click on ⚡ to see the export options - or you can always share this notebook's save file. (The file is pure Julia, by the way, and it's runnable! You'll learn more about this in the advanced introduction.)

Final notes

If anything about this introduction left you confused, something doesn't work, or you have a cool new idea - don't hesitate to contact us! You can do it right from this screen, using the `Instant feedback` form in the bottom right.

Also, if you were wondering where `confusing_function` came from, here you go! Remember that you, too, can place code wherever you like.

`Main.workspace3.confusing_function`

Have fun using Pluto!

~ Fons van der Plas & Nicholas Bochenski