Interview Cake

← course home (/table-of-contents)

# Short Circuit Evaluation

**Short-circuit evaluation** is a strategy most programming languages (including Python 3.6) use to avoid unnecessary work. For example, say we had a conditional like this:

Python 3.6 ▾

```
if it_is_friday and it_is_raining:
    print("board games at my place!")
```

Let's say `it_is_friday` is false. Because Python 3.6 short-circuits evaluation, it wouldn't bother checking the value of `it_is_raining`—it knows that either way the condition is false and we won't print the invitation to board game night.

We can use this to our advantage. For example, say we have a check like this:

Python 3.6 ▾

```
if friends['Becky'].is_free_this_friday():
    invite_to_board_game_night(friends['Becky'])
```

What happens if `'Becky'` isn't in our `friends` dictionary? We'll get a `KeyError` when we run `friends['Becky']`.

Instead, we could first confirm that Becky and I are still on good terms:

Python 3.6 ▾

```
if 'Becky' in friends and friends['Becky'].is_free_this_friday():
    invite_to_board_game_night(friends['Becky'])
```

This way, if `'Becky'` *isn't* in `friends`, Python will ignore the rest of the conditional and avoid throwing the `KeyError`.

> This is all hypothetical, of course. It's not like things with Becky are weird or anything. We're totally cool. She's still in my friends dictionary for sure and I hope I'm still in hers and Becky if you're reading this I just want you to know you're still in my friends dictionary.