ExUnit cheatsheet

Test cases

```
defmodule MyTest do
  use ExUnit.Case
  use ExUnit.Case, async: true  # for async

  test "the truth" do
    assert 1 + 1 == 2
  end
end
```

Capture logs

```
config :ex_unit, capture_logs: true
```

Assertions

```
assert x == y
refute x == y

assert_raise ArithmeticError, fn ->
  1 + "test"
end

assert_raise ArithmeticError, "message", fn -> ...
assert_raise ArithmeticError, ~r/message/, fn -> ...

flunk "This should've been an error"
```

Capture IO

```
import ExUnit.CaptureIO

test "capture io" do
  result = capture_io(fn ->
    IO.puts "sup"
  end)

  assert result == "sup\n"
end
```

Async

```
defmodule AssertionTest do
  # run concurrently with other test cases
  use ExUnit.Case, async: true
end
```

Setup

Pattern matching

```
setup do
  {:ok, name: "John"}
end
```

```
test "it works", %{name: name} do
  assert name == "John"
end
```

Setup

```
defp my_hook(_context) do
  # Invoked in every block in "a block"
  {:ok, name: "John", age: 54}
end

describe "a block" do
  setup [:my_hook]

  test "John's age", context do
    assert context[:name] == "John"
    assert context[:age] == 54
  end
end
```