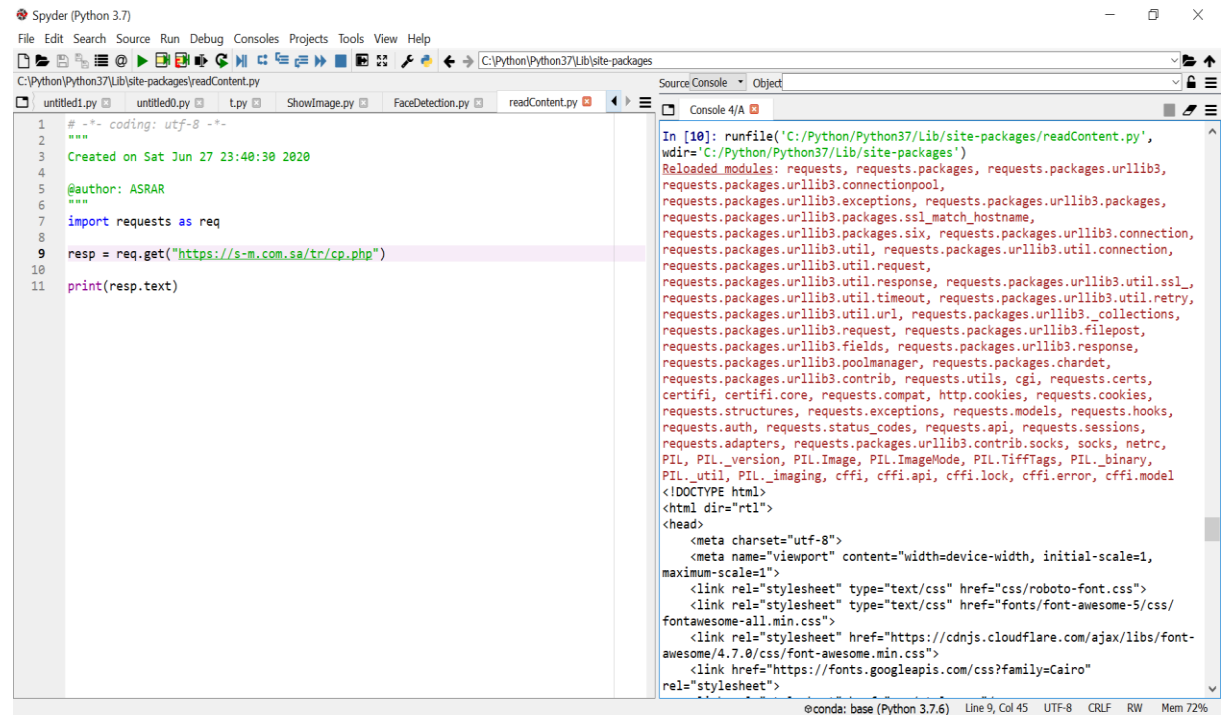


Task1: Read the content of a URL

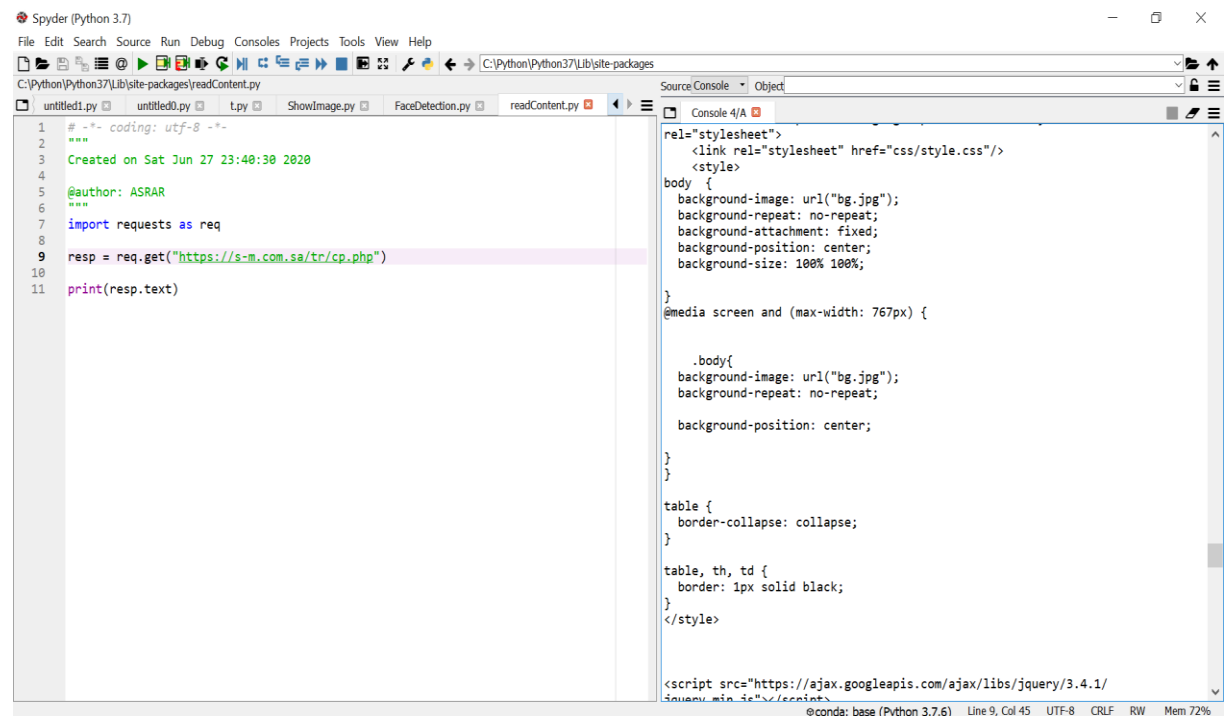
Code + Run



The screenshot shows the Spyder Python IDE interface. The editor window displays a Python script named `readContent.py` with the following code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Jun 27 23:40:30 2020
4
5 @author: ASRAR
6 """
7 import requests as req
8
9 resp = req.get("https://s-m.com.sa/tr/cp.php")
10
11 print(resp.text)
```

The console window on the right shows the output of the script, which is the HTML content of the URL `https://s-m.com.sa/tr/cp.php`. The output starts with `<!DOCTYPE html>` and includes a `<html>` tag with various attributes and a `<head>` section containing meta tags and links to stylesheets.



The screenshot shows the same Spyder Python IDE interface. The editor window displays the same Python script as in the previous screenshot. The console window on the right shows a different output, which appears to be a CSS stylesheet. The output includes a `<link>` tag for a stylesheet, a `<style>` tag with a `body` selector, and a `@media` query for screens with a maximum width of 767px. The `body` selector includes a `background-image` property set to `url("bg.jpg")` and a `background-position` property set to `center`. The `@media` query also includes a `background-image` property set to `url("bg.jpg")` and a `background-position` property set to `center`.

```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Python\Python37\Lib\site-packages\readContent.py
untitled1.py untitled0.py t.py ShowImage.py FaceDetection.py readContent.py
1 #-*- coding: utf-8 -*-
2 """
3 Created on Sat Jun 27 23:40:30 2020
4
5 @author: ASRAR
6 """
7 import requests as req
8
9 resp = req.get("https://s-m.com.sa/tr/cp.php")
10
11 print(resp.text)

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/
jquery.min.js"></script>

</head>

<body class="form-v5" >

  <div class="page-content" >
    <div class="form-v5-content" style="background-color:#e3fbff">
      <div class="form-detail">
        <a href="https://s-m.com.sa/"></
a>
      <h3 align='center' style='color:#85113d;'>كلمة المرور غير متطابقة</h3>

      <br><br>
    </div>
  </div>
</div>
</div>
</div>
<br>

  </div>
</div>
</div>

<script>
document.getElementById("phone").addEventListener("change", myFunction);
```

```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Python\Python37\Lib\site-packages\readContent.py
untitled1.py untitled0.py t.py ShowImage.py FaceDetection.py readContent.py
1 #-*- coding: utf-8 -*-
2 """
3 Created on Sat Jun 27 23:40:30 2020
4
5 @author: ASRAR
6 """
7 import requests as req
8
9 resp = req.get("https://s-m.com.sa/tr/cp.php")
10
11 print(resp.text)

function myFunction() {
var siv = document.getElementById("phone").value;

var res = siv.charAt(0)

var zero= "0";
var n = res.localeCompare(zero);
//alert(n);
if(n > 0 || siv.length < 10){
document.getElementById("phone").value = "";
alert("رقم الهاتف غير صحيح");
}

}
</script>

<script>
var x, i, j, selElmnt, a, b, c;
/*look for any elements with the class "custom-select":*/
x = document.getElementsByClassName("custom-select");
for (i = 0; i < x.length; i++) {
selElmnt = x[i].getElementsByTagName("select")[0];
/*for each element, create a new DIV that will act as the selected item:*/
a = document.createElement("DIV");
a.setAttribute("class", "select-selected");
a.innerHTML = selElmnt.options[selElmnt.selectedIndex].innerHTML;
x[i].appendChild(a);
/*for each element, create a new DIV that will contain the option list:*/
b = document.createElement("DIV");
```

conda: base (Python 3.7.6) Line 9, Col 45 UTF-8 CRLF RW Mem 72%

The screenshot shows the Spyder Python IDE interface. The left pane displays a Python script named `readContent.py` with the following content:

```
1 #-*- coding: utf-8 -*-
2 """
3 Created on Sat Jun 27 23:40:30 2020
4
5 @author: ASRAR
6 """
7 import requests as req
8
9 resp = req.get("https://s-m.com.sa/tr/cp.php")
10
11 print(resp.text)
```

The right pane shows the JavaScript code that is generated or associated with the Python script. It includes DOM manipulation and event handling logic:

```
c = document.createElement("DIV");
c.innerHTML = selElmnt.options[j].innerHTML;
c.addEventListener("click", function(e) {
    /*when an item is clicked, update the original select box,
    and the selected item:*/
    var y, i, k, s, h;
    s = this.parentNode.parentNode.getElementsByTagName("select")[0];
    h = this.parentNode.previousSibling;
    for (i = 0; i < s.length; i++) {
        if (s.options[i].innerHTML == this.innerHTML) {
            s.selectedIndex = i;
            h.innerHTML = this.innerHTML;
            y = this.parentNode.getElementsByTagName("same-as-selected");
            for (k = 0; k < y.length; k++) {
                y[k].removeAttribute("class");
            }
            this.setAttribute("class", "same-as-selected");
            break;
        }
    }
    h.click();
});
b.appendChild(c);
x[i].appendChild(b);
a.addEventListener("click", function(e) {
    /*when the select box is clicked, close any other select boxes,
    and open/close the current select box:*/
    e.stopPropagation();
    closeAllSelect(this);
    this.nextSibling.classList.toggle("select-hide");
    this.classList.toggle("select-arrow-active");
});
}
function closeAllSelect(elmnt) {
    /*a function that will close all select boxes in the document,
    except the current select box:*/
    var x, y, i, arrNo = [];
    x = document.getElementsByTagName("select-items");
    y = document.getElementsByTagName("select-selected");
    for (i = 0; i < y.length; i++) {
        if (elmnt == y[i]) {
            arrNo.push(i)
        } else {
            y[i].classList.remove("select-arrow-active");
        }
    }
    for (i = 0; i < x.length; i++) {
        if (arrNo.indexOf(i)) {
            x[i].classList.add("select-hide");
        }
    }
}
/*if the user clicks anywhere outside the select box,
then close all select boxes:*/
document.addEventListener("click", closeAllSelect);
</script>
</body>
</html>
```

This screenshot continues the view of the Spyder IDE, showing the final part of the JavaScript code in the right pane:

```
a.addEventListener("click", function(e) {
    /*when the select box is clicked, close any other select boxes,
    and open/close the current select box:*/
    e.stopPropagation();
    closeAllSelect(this);
    this.nextSibling.classList.toggle("select-hide");
    this.classList.toggle("select-arrow-active");
});
}
function closeAllSelect(elmnt) {
    /*a function that will close all select boxes in the document,
    except the current select box:*/
    var x, y, i, arrNo = [];
    x = document.getElementsByTagName("select-items");
    y = document.getElementsByTagName("select-selected");
    for (i = 0; i < y.length; i++) {
        if (elmnt == y[i]) {
            arrNo.push(i)
        } else {
            y[i].classList.remove("select-arrow-active");
        }
    }
    for (i = 0; i < x.length; i++) {
        if (arrNo.indexOf(i)) {
            x[i].classList.add("select-hide");
        }
    }
}
/*if the user clicks anywhere outside the select box,
then close all select boxes:*/
document.addEventListener("click", closeAllSelect);
</script>
</body>
</html>
```

Reference:

- 1- <https://www.datacamp.com/community/tutorials/making-http-requests-in-python>
- 2- <http://zetcode.com/python/requests/>