In [1]:
```
# print statements
variable = "right in the strings!"
print(f"f strings allow you to embed variables {variable}")
```

```
f strings allow you to embed variables right in the strings!
```

In [2]:
```
#This is  a 'Code' Cell
print("This is  code cell")
```

```
This is  code cell
```

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('./deeplearning.mplstyle')
```

```
Matplotlib is building the font cache; this may take a moment.
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
File E:\anacoda11\Lib\site-packages\matplotlib\style\core.py:166, in use(style)
    165 try:
--> 166     style = _rc_params_in_file(style)
    167 except IOError as err:

File E:\anacoda11\Lib\site-packages\matplotlib\__init__.py:850, in _rc_params_in_file(fname, transform, fail_on_error)
    849 rc_temp = {}
--> 850 with _open_file_or_url(fname) as fd:
    851     try:

File E:\anacoda11\Lib\contextlib.py:137, in _GeneratorContextManager.__enter__(self)
    136 try:
--> 137     return next(self.gen)
    138 except StopIteration:

File E:\anacoda11\Lib\site-packages\matplotlib\__init__.py:827, in _open_file_or_url(fname)
    826 fname = os.path.expanduser(fname)
--> 827 with open(fname, encoding='utf-8') as f:
    828     yield f

FileNotFoundError: [Errno 2] No such file or directory: './deeplearning.mplstyle'

The above exception was the direct cause of the following exception:

OSError                                   Traceback (most recent call last)
Cell In[3], line 3
      1 import numpy as np
      2 import matplotlib.pyplot as plt
----> 3 plt.style.use('./deeplearning.mplstyle')

File E:\anacoda11\Lib\site-packages\matplotlib\style\core.py:168, in use(style)
    166         style = _rc_params_in_file(style)
    167     except IOError as err:
--> 168         raise IOError(
    169             f"{style!r} is not a valid package style, path of style "
    170             f"file, URL of style file, or library style name (library "
    171             f"styles are listed in `style.available`)") from err
    172 filtered = {}
    173 for k in style:  # don't trigger RcParams.__getitem__('backend')

OSError: './deeplearning.mplstyle' is not a valid package style, path of style file, URL of style file, or library style name (library styles are listed in `style.available`)
```

In [4]:
```
# x_train is the input variable (size in 1000 square feet)
# y_train is the target (price in 1000s of dollars)
x_train = np.array([1.0, 2.0])
y_train = np.array([300.0, 500.0])
print(f"x_train = {x_train}")
print(f"y_train = {y_train}")
```

```
x_train = [1. 2.]
y_train = [300. 500.]
```

In [5]:
```
# m is the number of training examples
print(f"x_train.shape: {x_train.shape}")
m = x_train.shape[0]
print(f"Number of training examples is: {m}")
```

```
x_train.shape: (2,)
Number of training examples is: 2
```

In [6]:
```
# m is the number of training examples
m = len(x_train)
print(f"Number of training examples is: {m}")
```

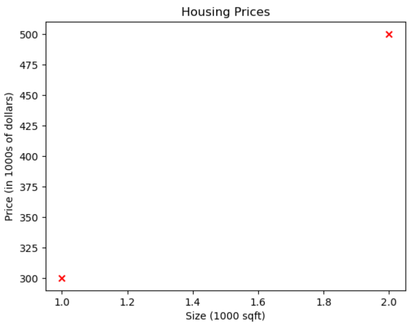
```
Number of training examples is: 2
```

In [7]:
```
i = 1 # Change this to 1 to see (x^1, y^1)

x_i = x_train[i]
y_i = y_train[i]
print(f"(x^({i}), y^({i})) = ({x_i}, {y_i})")
```

```
(x^(1), y^(1)) = (2.0, 500.0)
```

In [8]:
```
# Plot the data points
plt.scatter(x_train, y_train, marker='x', c='r')
# Set the title
plt.title("Housing Prices")
# Set the y-axis label
plt.ylabel('Price (in 1000s of dollars)')
# Set the x-axis label
plt.xlabel('Size (1000 sqft)')
plt.show()
```



In [9]:
```
w = 200
b = 100
print(f"w: {w}")
print(f"b: {b}")
```

```
w: 200
b: 100
```

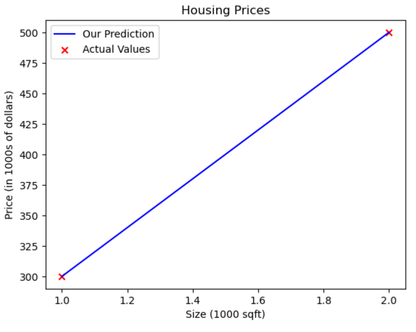
In [10]:
```
def compute_model_output(x, w, b):
    """
    Computes the prediction of a linear model
    Args:
      x (ndarray (m,)): Data, m examples
      w,b (scalar)    : model parameters
    Returns
      y (ndarray (m,)): target values
    """
    m = x.shape[0]
    f_wb = np.zeros(m)
    for i in range(m):
        f_wb[i] = w * x[i] + b
    return f_wb
```

In [11]:
```
tmp_f_wb = compute_model_output(x_train, w, b,)


# Plot our model prediction
plt.plot(x_train, tmp_f_wb, c='b',label='Our Prediction')

# Plot the data points
plt.scatter(x_train, y_train, marker='x', c='r',label='Actual Values')

# Set the title
plt.title("Housing Prices")
# Set the y-axis label
plt.ylabel('Price (in 1000s of dollars)')
# Set the x-axis label
plt.xlabel('Size (1000 sqft)')
plt.legend()
plt.show()
```



In [12]:
```
w = 200
b = 100
x_i = 1.2
cost_1200sqft = w * x_i + b

print(f"${cost_1200sqft:.0f} thousand dollars")
```

```
$340 thousand dollars
```

In [13]:
```
f_wb = np.zeros(2)
print(f_wb)
```

```
[0. 0.]
```

In [ ]: