Recap from Monday's lecture

In [73]:
```
heights.head(7)
```

Out[73]:

	father	mother	gender	child
0	78.5	67.0	male	NaN
1	78.5	67.0	female	69.2
2	78.5	67.0	female	69.0
3	78.5	67.0	female	69.0
4	75.5	66.5	male	NaN
5	75.5	66.5	male	NaN
6	75.5	66.5	female	NaN

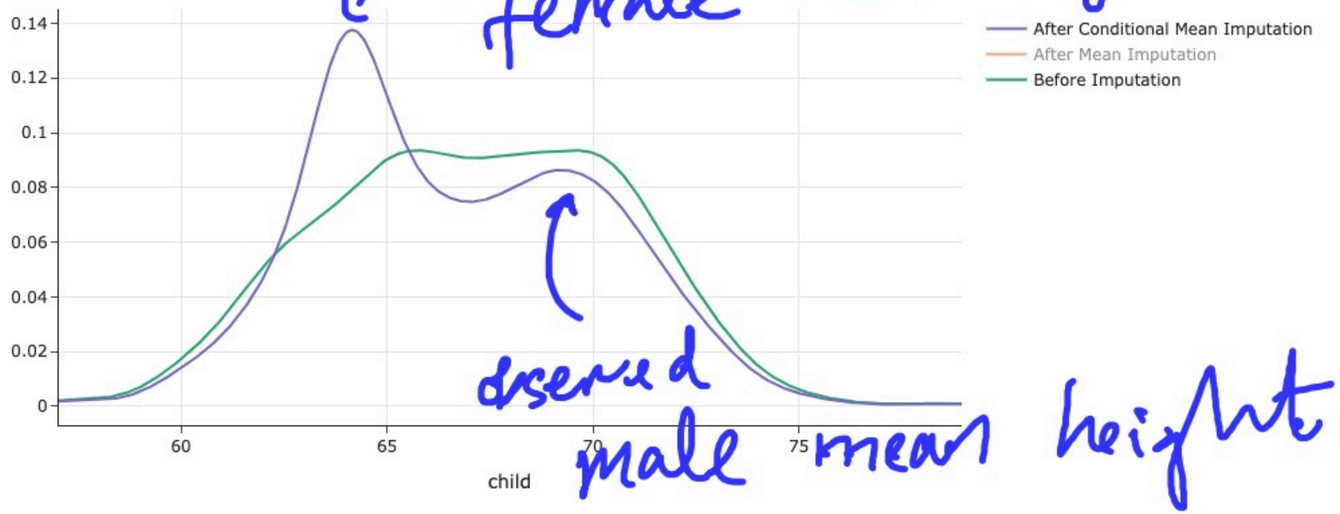
1) Fill in (impute) with the mean of the observed values

2) Conditional mean imputation dependent on gender

Pros and cons of conditional mean imputation

- Instead of having a single "spike", the conditionally-imputed distribution has two smaller "spikes".
  In this case, one at the observed `'female'` mean and one at the observed `'male'` mean.

In [68]:
```
multiple_kdes({'Before Imputation': heights['child'],
               'After Mean Imputation': mean_imputed,
               'After Conditional Mean Imputation': conditional_mean_imputed})
```



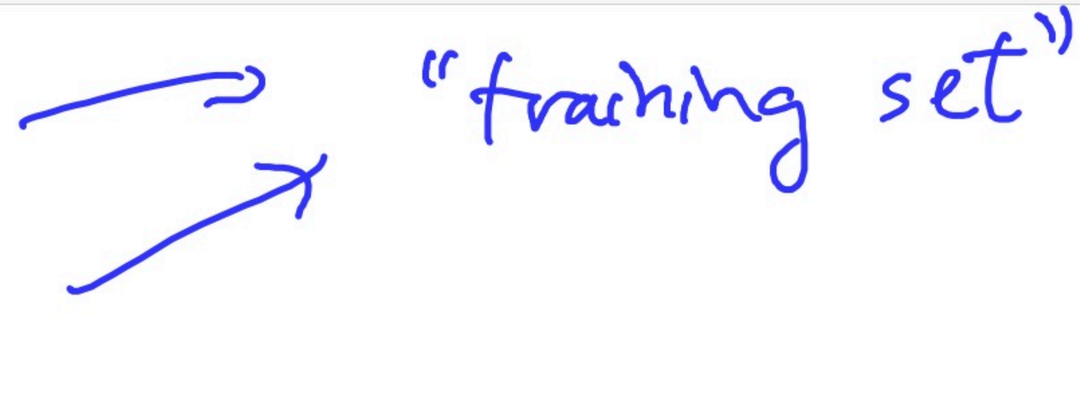
Idea: Regression imputation

- A common solution is to fill in missing values by using other features to **predict** what the missing value would have been.

In [78]:
```
# There's nothing special about the values passed into .iloc below;
# they're just for illustration.
heights.iloc[[0, 2, 919, 11, 4, 8, 9]]
```

Out[78]:

	father	mother	gender	child
0	78.5	67.0	male	NaN
2	78.5	67.0	female	69.0
919	64.0	64.0	female	NaN
11	75.0	64.0	male	68.5
4	75.5	66.5	male	NaN
8	75.0	64.0	male	71.0
9	75.0	64.0	female	68.0



Idea: Probabilistic imputation

5 13 12 13 7 4 4

- Since **we don't know** what the missing values would have been, one could argue our technique for filling in missing values should incorporate this uncertainty.

- We could fill in missing values using a **random sample** of observed values.

  This avoids the key issue with mean imputation, where we fill all missing values with the same one value. It also limits the bias present if the missing values weren't a representative sample, since we're filling them in with a range of different values.

In [ ]:
```python
# impute_prob should take in a Series with missing values and return an imputed Series.
def impute_prob(s):
    s = s.copy()
    # Find the number of missing values.
    num_missing = s.isna().sum()
    # Take a sample of size num_missing from the present values.
    sample = np.random.choice(s.dropna(), num_missing)
    # Fill in the missing values with our random sample.
    s.loc[s.isna()] = sample
    return s
```

implementation.

```
    num_missing = s.isna().sum()
    # Take a sample of size num_missing from the present values.
    sample = np.random.choice(s.dropna(), num_missing)
    # Fill in the missing values with our random sample.
    s.loc[s.isna()] = sample
    return s
```

- Each time we run the cell below, the missing values in `heights['child']` are filled in with a different sample of present values in `heights['child']`!

In [96]:
```
# The number at the very top is constantly changing!
prob_imputed = impute_prob(heights['child'])
print('Mean:', prob_imputed.mean())
prob_imputed
```

```
Mean: 67.1041755888651
```

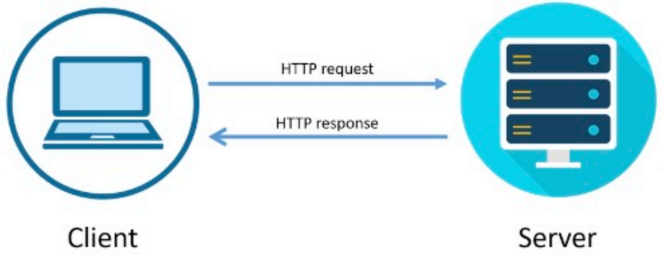
Out[96]:
```
0      68.0
1      69.2
2      69.0
       ...
931    61.0
932    66.5
933    57.0
Name: child, Length: 934, dtype: float64
```

originally missing!

- To account for the fact that each run is slightly different, a common strategy is **multiple imputation**.

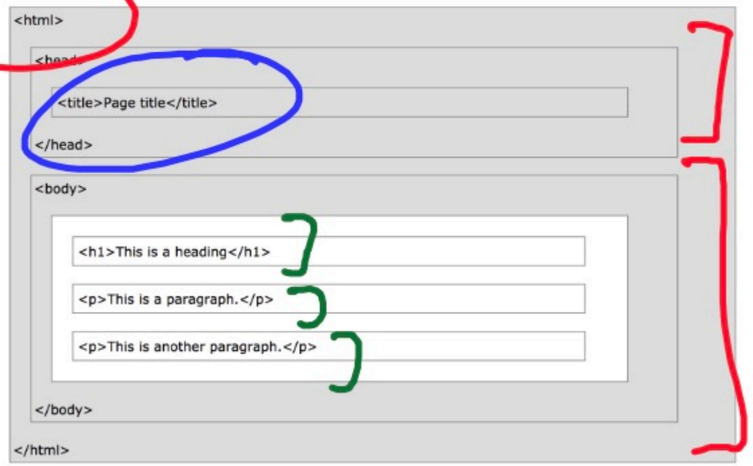
- HTTP stands for **Hypertext Transfer Protocol**.
  It was developed in 1989 by Tim Berners-Lee (and friends). The "S" in HTTPS stands for "secure".

- HTTP follows the **request-response** model, in which a **request** is made by the **client** and a **response** is returned by the **server**.

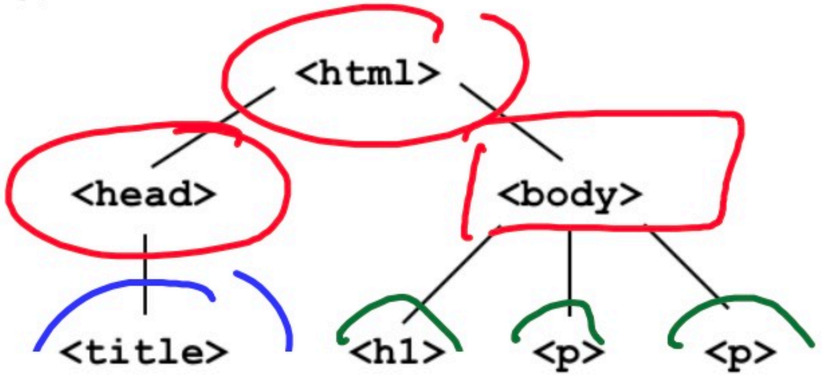


- **Example**: YouTube search 🎥.
  - Consider the following URL: https://www.youtube.com/results?search_query=luka+lakers+trade.
  - Your web browser, a **client**, makes an HTTP **request** with a search query.
  - The **server**, YouTube, is a computer that is sitting somewhere else.
  - The server returns a **response** that contains the search results.
  - **Note**: ?search_query=luka+lakers+trade is called a "query string."

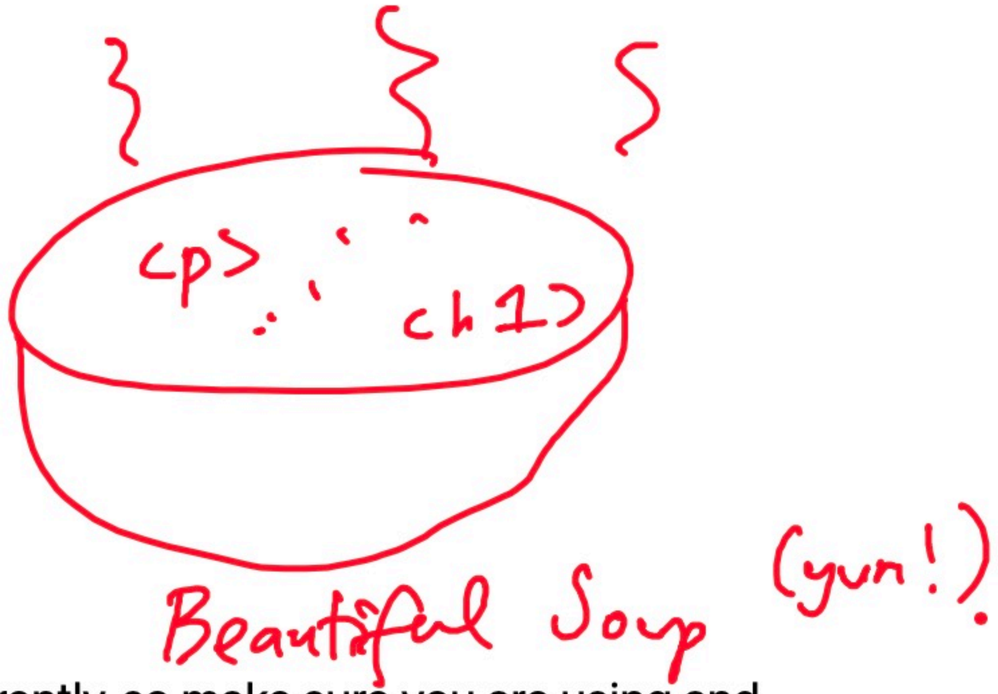
- **HTML document**: The totality of markup that makes up a webpage.



- **Document Object Model (DOM)**: The internal representation of an HTML document as a hierarchical **tree** structure.



Beautiful Soup 🍜



- Beautiful Soup 4 is a Python HTML parser.
  Remember, **parse** means to "extract meaning from a sequence of symbols".

- **Warning**: Beautiful Soup 4 and Beautiful Soup 3 work differently, so make sure you are using and looking at documentation for Beautiful Soup 4.
  Rest assured, the `pds` conda environment already has Beautiful Soup 4 installed.

Finding elements in a BeautifulSoup object

- The two main methods you will use to extract information from a BeautifulSoup object are `find` and `find_all`.

- `soup.find(tag)` finds the **first** instance of a tag (the first one on the page, i.e. the first one that DFS sees), and returns just that tag.
  It has several optional arguments, including some that involve defining `lambda` functions: **look at the documentation!**

- `soup.find_all(tag)` will find **all** instances of a tag, and returns a **list** of tags.

- Remember: **`find` finds tags!**

Out[22]:
```
<h1>Heading here</h1>
<p>My First paragraph</p>
<p>My <em>second</em> paragraph</p>
<hr/>
</div>
```

- Let's try and find the `<div>` element that has an `id` attribute equal to `'nav'`.

In [ ]:

In [23]:
```
soup
```

Out[23]:
```
<html>
<body>
<div id="content">
<h1>Heading here</h1>
<p>My First paragraph</p>
<p>My <em>second</em> paragraph</p>
<hr/>
</div>
<div id="nav">
<ul>
<li>item 1</li>
<li>item 2</li>
<li>item 3</li>
</ul>
</div>
</body>
</html>
```

- Let's try and find the `<div>` element that has an `id` attribute equal to `'nav'`.

In [24]:
```
soup.find('div', attrs={'id': 'nav'})
```

Out[24]:
```
<div id="nav">
<ul>
<li>item 1</li>
<li>item 2</li>
<li>item 3</li>
</ul>
</div>
```

In [23]:
```
soup
```

Out[23]:
```
<html>
<body>
<div id="content">
<h1>Heading here</h1>
<p>My First paragraph</p>
<p>My <em>second</em> paragraph</p>
<hr/>
</div>
<div id="nav">
<ul>
<li>item 1</li>
<li>item 2</li>
<li>item 3</li>
</ul>
</div>
</body>
</html>
```

Tags: simplicity understand

"You may not be her first, her last, or her only. She loved before she may love again. But if she loves you now, what else matters? She's not perfect—you aren't either, and the two of you may never be perfect together but if she can make you laugh, cause you to think twice, and admit to being human and making mistakes, hold onto her and give her the most you can. She may not be thinking about you every second of the day, but she will give you a part of her that she knows you can break—her heart. So don't hurt her, don't change her, don't analyze and don't expect more than she can give. Smile when she makes you happy, let her know when she makes you mad, and miss her when she's not there."

by Bob Marley (about)

Tags: love

"I like nonsense, it wakes up the brain cells. Fantasy is a necessary ingredient in living."

by Dr. Seuss (about)

Tags: fantasy

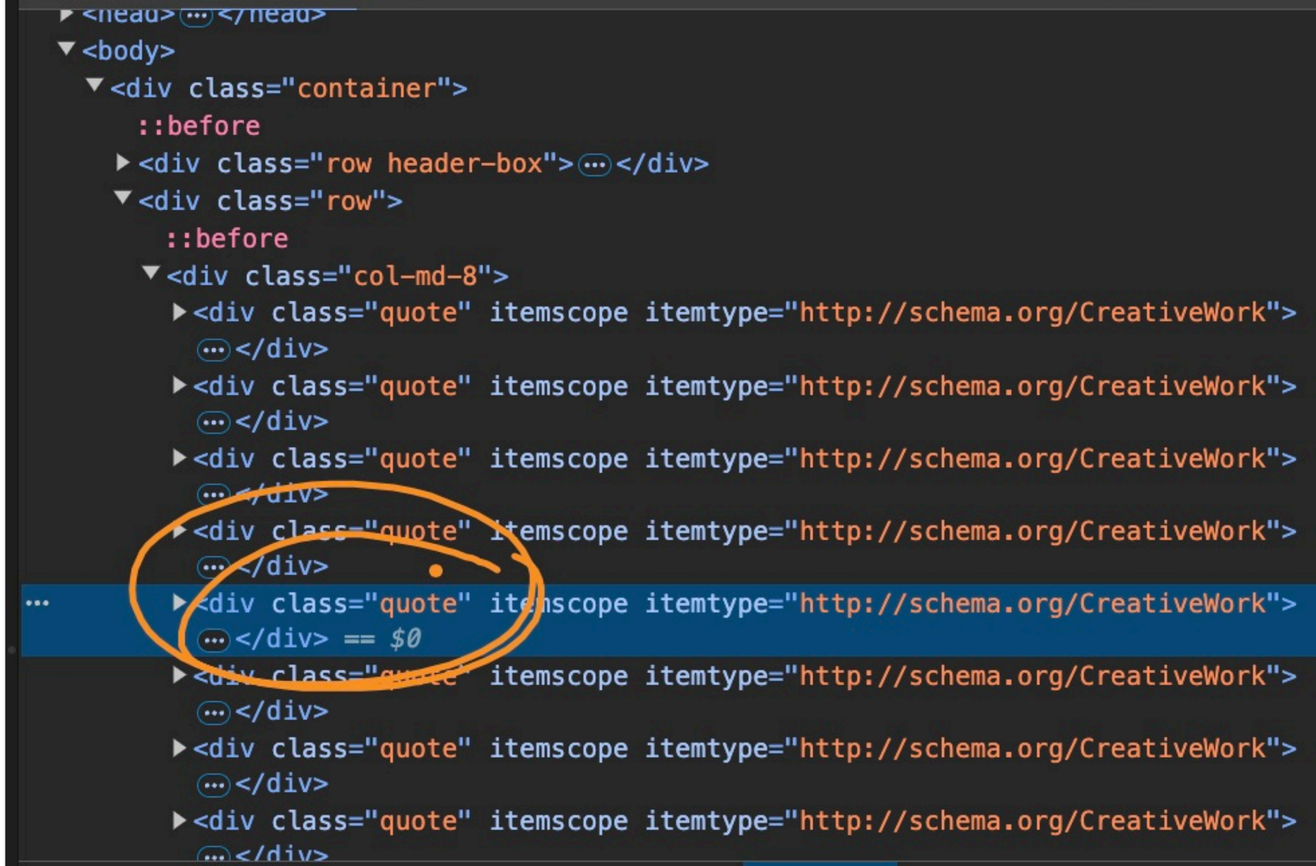
"I may not have gone where I intended to go, but I think I have ended up where I needed to be."

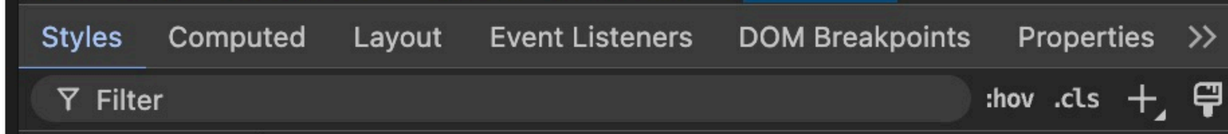
by Douglas Adams (about)

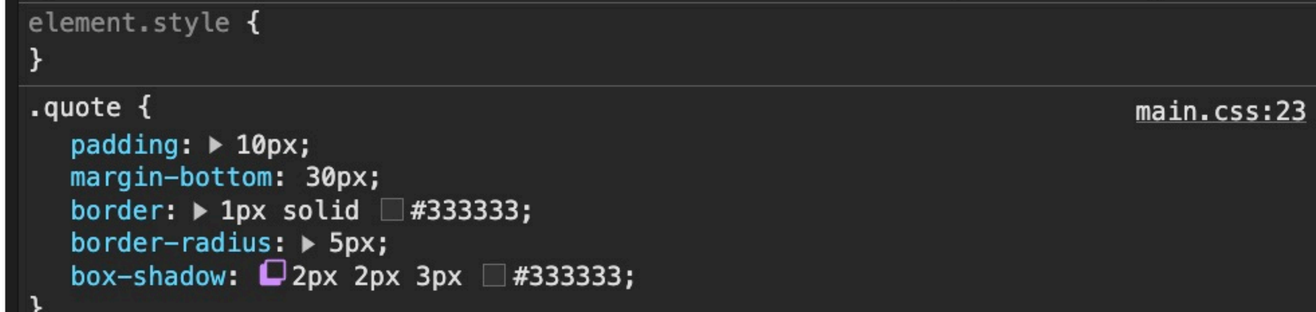
Tags: life navigation

"The opposite of love is not hate, it's indifference. The opposite of art is not ugliness, it's indifference. The opposite of faith is not heresy, it's indifference. And the opposite of life is not death, it's indifference."

by Elie Wiesel (about)







```
# len(divs)
divs = soup.find_all('div', class_='quote')
```

In [59]:
```
divs[0]
```

Out[59]:
```
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">“This life is what you make it. No matter what, you're going t
o mess up sometimes, it's a universal truth. But the good part is you get to decide how you're go
ing to mess it up. Girls will be your friends - they'll act like it anyway. But just remember, so
me come, some go. The ones that stay with you through everything - they're your true best friend
s. Don't let go of them. Also remember, sisters make the best friends in the world. As for lover
s, well, they'll come and go too. And baby, I hate to say it, most of them - actually pretty much
all of them are going to break your heart, but you can't give up because if you give up, you'll n
ever find your soulmate. You'll never find that half who makes you whole and that goes for everyt
hing. Just because you fail once, doesn't mean you're gonna fail at everything. Keep trying, hold
on, and always, always, always believe in yourself, because if you don't, then who will, sweetie?
So keep your head high, keep your chin up, and most importantly, keep smiling, because life's a b
eautiful thing and there's so much to smile about.”</span>
<span>by <small class="author" itemprop="author">Marilyn Monroe</small>
<a href="/author/Marilyn-Monroe">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="friends,heartbreak,inspirational,life,love,sisters" i
temprop="keywords"/>
<a class="tag" href="/tag/friends/page/1/">friends</a>
<a class="tag" href="/tag/heartbreak/page/1/">heartbreak</a>
<a class="tag" href="/tag/inspirational/page/1/">inspirational</a>
<a class="tag" href="/tag/life/page/1/">life</a>
<a class="tag" href="/tag/love/page/1/">love</a>
<a class="tag" href="/tag/sisters/page/1/">sisters</a>
```

In [66]:
```
# The URL for the author.
divs[0]
```

Out[66]:
```
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">“This life is what you make it. No matter what, you're going t
o mess up sometimes, it's a universal truth. But the good part is you get to decide how you're go
ing to mess it up. Girls will be your friends - they'll act like it anyway. But just remember, so
me come, some go. The ones that stay with you through everything - they're your true best friend
s. Don't let go of them. Also remember, sisters make the best friends in the world. As for lover
s, well, they'll come and go too. And baby, I hate to say it, most of them - actually pretty much
all of them are going to break your heart, but you can't give up because if you give up, you'll n
ever find your soulmate. You'll never find that half who makes you whole and that goes for everyt
hing. Just because you fail once, doesn't mean you're gonna fail at everything. Keep trying, hold
on, and always, always, always believe in yourself, because if you don't, then who will, sweetie?
So keep your head high, keep your chin up, and most importantly, keep smiling, because life's a b
eautiful thing and there's so much to smile about.”</span>
<span>by <small class="author" itemprop="author">Marilyn Monroe</small>
<a href="/author/Marilyn-Monroe">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="friends,heartbreak,inspirational,life,love,sisters" i
temprop="keywords"/>
<a class="tag" href="/tag/friends/page/1/">friends</a>
<a class="tag" href="/tag/heartbreak/page/1/">heartbreak</a>
<a class="tag" href="/tag/inspirational/page/1/">inspirational</a>
<a class="tag" href="/tag/life/page/1/">life</a>
<a class="tag" href="/tag/love/page/1/">love</a>
<a class="tag" href="/tag/sisters/page/1/">sisters</a>
</div>
</div>
```

In [73]:
```
# The quote's tags.
divs[0].
```

find("meta", class_ = "keywords").get("content")

Out[73]:
```
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">“This life is what you make it. No matter what, you're going t
o mess up sometimes, it's a universal truth. But the good part is you get to decide how you're go
ing to mess it up. Girls will be your friends - they'll act like it anyway. But just remember, so
me come, some go. The ones that stay with you through everything - they're your true best friend
s. Don't let go of them. Also remember, sisters make the best friends in the world. As for lover
s, well, they'll come and go too. And baby, I hate to say it, most of them - actually pretty much
all of them are going to break your heart, but you can't give up because if you give up, you'll n
ever find your soulmate. You'll never find that half who makes you whole and that goes for everyt
hing. Just because you fail once, doesn't mean you're gonna fail at everything. Keep trying, hold
on, and always, always, always believe in yourself, because if you don't, then who will, sweetie?
So keep your head high, keep your chin up, and most importantly, keep smiling, because life's a b
eautiful thing and there's so much to smile about.”</span>
<span>by <small class="author" itemprop="author">Marilyn Monroe</small>
<a href="/author/Marilyn-Monroe">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="friends,heartbreak,inspirational,life,love,sisters" i
temprop="keywords"/>
<a class="tag" href="/tag/friends/page/1/">friends</a>
<a class="tag" href="/tag/heartbreak/page/1/">heartbreak</a>
<a class="tag" href="/tag/inspirational/page/1/">inspirational</a>
<a class="tag" href="/tag/life/page/1/">life</a>
<a class="tag" href="/tag/love/page/1/">love</a>
<a class="tag" href="/tag/sisters/page/1/">sisters</a>
</div>
</div>
```

- Remember, the 200 status code is good! Let's take a look at the text, the same way we did before:

curly brackets!

In [90]:
```
res.text
```


```
0,"version":{"name":"soulsilver","url":"https://pokeapi.co/api/v2/version/1
6/"}},{"rarity":50,"version":{"name":"black","url":"https://pokeapi.co/api/v
2/version/17/"}},{"rarity":50,"version":{"name":"white","url":"https://pokeap
i.co/api/v2/version/18/"}}]},{"item":{"name":"light-ball","url":"https://poke
api.co/api/v2/item/213/"},"version_details":[{"rarity":5,"version":{"name":"r
uby","url":"https://pokeapi.co/api/v2/version/7/"}},{"rarity":5,"version":{"n
ame":"sapphire","url":"https://pokeapi.co/api/v2/version/8/"}},{"rarity":5,"v
ersion":{"name":"emerald","url":"https://pokeapi.co/api/v2/version/9/"}},{"ra
rity":5,"version":{"name":"diamond","url":"https://pokeapi.co/api/v2/version/
12/"}},{"rarity":5,"version":{"name":"pearl","url":"https://pokeapi.co/api/v
2/version/13/"}},{"rarity":5,"version":{"name":"platinum","url":"https://poke
api.co/api/v2/version/14/"}},{"rarity":5,"version":{"name":"heartgold","ur
l":"https://pokeapi.co/api/v2/version/15/"}},{"rarity":5,"version":{"name":"s
oulsilver","url":"https://pokeapi.co/api/v2/version/16/"}},{"rarity":1,"versi
on":{"name":"black","url":"https://pokeapi.co/api/v2/version/17/"}},{"rarit
y":1,"version":{"name":"white","url":"https://pokeapi.co/api/v2/version/1
8/"}},{"rarity":5,"version":{"name":"black-2","url":"https://pokeapi.co/api/v
```