I had a problem while doing this assignment. I could not find any proxy server that allowed bet365.com traffic to pass.
So I tested my solution on 3 other websites.

1. One problem I was tackling with during this task, was finding the target link website containing win outrights for World Cup 2018.

The way I finally did it, is finding and following websites via links related to the topic of World Cup.
Since the task was related only to World Cup, this method as it is (without changing the link titles) cannot be applied to any problem, out of the box.

Another idea that I had, was using the methodology called BM25F, for scoring documents against a input query (finding similarity between query and document)
While this method could be applied to any potential query, it requires long time processing before doing and document scoring.
Because Python's library request does not support asynchronous JavaScript communication, I had to use Selenium for aforementioned preprocessing, which of course implies long processing time.
Yet again, since the task concerns World Cup 2018, this kind of method sounds like too much for this kind of problem.

Then I figured, that document scoring is exactly what Google is doing, and I decided to try to use its results to find the target link ('*google.py*'). However Google's first result for an example query 'https://paddypower.com world cup 2018 win outrights' was not exactly what I was expecting it to be.

My method is in *link_finder.py*

2. Another problem I had, was extracting correct numbers from website.
Again, using Selenium, I looked for a parent node of node containing the team's name.
If the parent has no nodes containing score in a form *number/number*, I moved to it's grandparent etc…