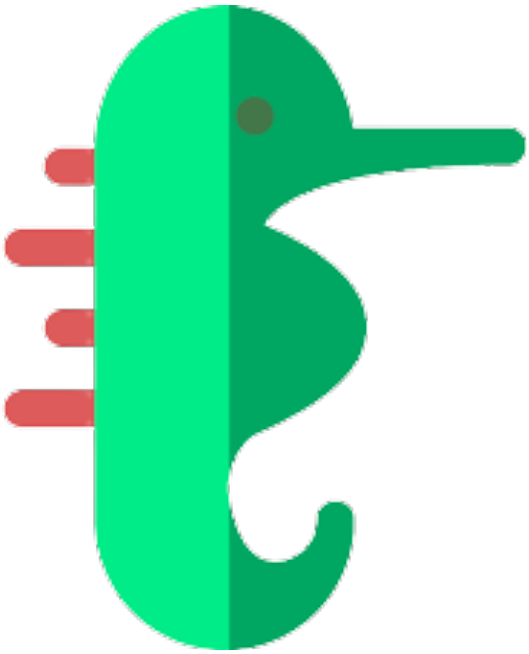


# SeadragonSearch Analytics Documentation

**None**

*Dominic Cain*

*None*

# Table of contents

# 1. Overview of the Project

# 2. Extensions Installed Overview

This is a VERY VERY small overview to what you can do with this. I will just highlight some of them, because those are the only documentation syntax that I commonly use and usually remember.

## 2.1 Admonitions

These are kind of those fancy boxes that you usually in cool Science Books that adds extra information.

**Note**

As you can see this box, is very attractive.

The syntax for this is:

```
!!! note
As you can see this box, is very attractive.
```

**What If You want a different Title**

The syntax for this is:

```
!!! note "What If You want a different Title"
As you can see this box, is very attractive.
```

### 2.1.1 Icons

More info here

You can also change these icons by changing the first word after `!!!` or `???`.

`note`, `seealso`

**Note**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`abstract`, `summary`, `tldr`

**Abstract**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`info` , `todo`

**Info**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`tip` , `hint` , `important`

**Tip**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`success` , `check` , `done`

**Success**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`question` , `help` , `faq`

**Question**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`warning` , `caution` , `attention`

**Warning**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`failure` , `fail` , `missing`

**Failure**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`danger` , `error`

**Danger**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`bug`

**Bug**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`example`

**Example**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

`quote`, `cite`

**Quote**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

### 2.1.2 Collapsible Block

More info here

If things are getting a little bit crowded, why not make some of them collapsible?

**Example of a More Complex Documentation**

Here is the basic idea of bubble sort!

```
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```

**The Syntax for the Example Above**

````
    ??? Example "Example of a More Complex Documentation"
    Here is the basic idea of bubble sort!
    ```python
    def bubble_sort(items):
        for i in range(len(items)):
            for j in range(len(items) - 1 - i):
                if items[j] > items[j + 1]:
                    items[j], items[j + 1] = items[j + 1], items[j]
    ```
````

## 2.2 Code Highlight

This is powered by codehilite. Whenever, you need code, this is the one that makes it pretty.

*For example:*

```
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```

**Syntax of the Example Above**

```
    ``` python linenums="1"
    def bubble_sort(items):
        for i in range(len(items)):
            for j in range(len(items) - 1 - i):
                if items[j] > items[j + 1]:
                    items[j], items[j + 1] = items[j + 1], items[j]
    ```
```

### 2.2.1 Highlight Specific Code Lines

What if I want to show some cool lines? I could highlight which specific line number should be highlighted.

```
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```

**Syntax of the Example Above**

```
    ``` python hl_lines="2 3"
    def bubble_sort(items):
        for i in range(len(items)):
            for j in range(len(items) - 1 - i):
                if items[j] > items[j + 1]:
                    items[j], items[j + 1] = items[j + 1], items[j]
    ```
```

## 2.3 Latex / Math Symbol Renderer

This is for math nerds that needs some Maths in their documentation. More info on Latex here.

For example, the Pythagoras Theorem $$ a^2 + b^2 = c^2 $$

**Syntax of the Example Above**

```
$$ a^2 + b^2 = c^2 $$
```

### 2.3.1 Inline Latex

According to the results with the p-value $p < 0.05$, it means that we will reject the null Hypothesis $H_0$, and that there is a significant difference in the means.

## 2.4 Footnotes

*Woah woah woah! Getting a little bit nerdy referencer here!*

"You can tell that I don't know much about referencing"[1]. If you click this shiny number, it takes you to the bottom of the page where the reference is.

**Syntax of the Example Above**

```
"You can tell that I don't know much about referencing"[^1]

[^1]:
Book of Wisdom - John Doe
```

## 2.5 Content Tabs

Very useful for when you need one or the other.

For example, when dealing with multiple programming languages.

C

```c
#include <stdio.h>

int main(void) {
  printf("Hello world!\n");
  return 0;
}
```

C++

```cpp
#include <iostream>

int main(void) {
  std::cout << "Hello world!" << std::endl;
  return 0;
}
```

**Syntax of Above**

```
    === "C"

        ``` c
        #include <stdio.h>

        int main(void) {
        printf("Hello world!\n");
        return 0;
        }
        ```

    === "C++"

        ``` c++
        #include <iostream>

        int main(void) {
        std::cout << "Hello world!" << std::endl;
        return 0;
        }
        ```
```

## 2.6 Icons and Emoji

Just worth mentioning, not too sure if you're going to use it.

- – `.icons/material/account-circle.svg`
- :fontawesome-regular-laugh-wink: – `.icons/fontawesome/regular/laugh-wink.svg`
- :octicons-octoface-16: – `.icons/octicons/octoface-16.svg`

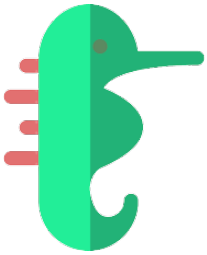
**Syntax of Above**

```
- :material-account-circle: – `.icons/material/account-circle.svg`
- :fontawesome-regular-laugh-wink: – `.icons/fontawesome/regular/laugh-wink.svg`
- :octicons-octoface-16: – `.icons/octicons/octoface-16.svg`
```

## 2.7 Images

Can be done with Markdown or HTML.

### 2.7.1 Image Captioning



*The Logo that Daphne from Coders for Causes gave me*

**Syntax of Above**

```
<figure>
    <img src="../images/seadragon.png" width="100" />
    <figcaption>The Logo that Daphne from Coders for Causes gave me</figcaption>
</figure>
```

### 2.7.2 Image Alignment


This is for when you have paragraphs and some text, but you really wanted those fancy images on the side. You can either say `left` or `right`. Now Let me just fill this with some random words so that the image doesn't look to weird.

**Syntax Above**

```
![Placeholder](https://dummyimage.com/600x400/f5f5f5/aaaaaa&text=-%20Image%20-){: align=left width=300 }

This is for when you have paragraphs and some text, but you really wanted those fancy images on the side. You can either say `left` or `right`. Now Let me
just fill this with some random words so that the image doesn't look to weird.
```

## 2.8 Graph In Markdown / Mermaid Markdown

More Information here.

What if you really just want to create some fancy graphs, but you really can't be bothered to:

1. Load some other software
2. Draw this graph that you wanted to show
3. Save this graph that you want to show
4. Upload this graph somewhere
5. Link this image back to this documentation

Like there are just soooo many steps.

Introducing **mermaid markdown**.

graph TD A --> B & C B --> C

**Syntax for Above**

```
```mermaid
graph TD
    A --> B & C
    B --> C
```
```

How about more complex ones? Is this complex enough for your

graph TD A[Hard] -->|Text| B(Round) B --> C{Decision} C -->|One| D[Result 1] C -->|Two| E[Result 2]

**Syntax for Above**

```
```mermaid
graph TD
    A[Hard] -->|Text| B(Round)
    B --> C{Decision}
    C -->|One| D[Result 1]
    C -->|Two| E[Result 2]
```
```

## 2.8.1 Some Examples of Other Charts

**Sequence Diagram**

**Result**

sequenceDiagram participant Alice participant Bob Alice->>John: Hello John, how are you? loop Healthcheck John->>John: Fight against hypochondria end Note right of John: Rational thoughts
prevail! John-->>Alice: Great! John->>Bob: How about you? Bob-->>John: Jolly good!

**Syntax**

```
```mermaid
sequenceDiagram
participant Alice
participant Bob
Alice->>John: Hello John, how are you?
loop Healthcheck
    John->>John: Fight against hypochondria
end
Note right of John: Rational thoughts <br/>prevail!
John-->>Alice: Great!
John->>Bob: How about you?
Bob-->>John: Jolly good!
```
```

#### Gantt Chart

**Result**

gantt dateFormat YYYY-MM-DD title Adding GANTT diagram to mermaid excludes weekdays 2014-01-10 section A section Completed task :done, des1, 2014-01-06,2014-01-08 Active task :active, des2, 2014-01-09, 3d Future task : des3, after des2, 5d Future task2 : des4, after des3, 5d

**Syntax**

```
```mermaid
gantt
dateFormat  YYYY-MM-DD
title Adding GANTT diagram to mermaid
excludes weekdays 2014-01-10

section A section
Completed task            :done,    des1, 2014-01-06,2014-01-08
Active task               :active,  des2, 2014-01-09, 3d
Future task               :         des3, after des2, 5d
Future task2               :         des4, after des3, 5d
```
```

#### Class Diagram

**Result**

classDiagram Class01 <|-- AveryLongClass : Cool Class03 *-- Class04 Class05 o-- Class06 Class07 .. Class08 Class09 --> C2 : Where am i? Class09 --* C3 Class09 --|> Class07 Class07 : equals() Class07 : Object[] elementData Class01 : size() Class01 : int chimp Class01 : int gorilla Class08 <--> C2: Cool label

**Syntax**

```
```mermaid
classDiagram
Class01 <|-- AveryLongClass : Cool
Class03 *-- Class04
Class05 o-- Class06
Class07 .. Class08
Class09 --> C2 : Where am i?
Class09 --* C3
Class09 --|> Class07
Class07 : equals()
Class07 : Object[] elementData
Class01 : size()
Class01 : int chimp
Class01 : int gorilla
Class08 <--> C2: Cool label
```
```

---

1. Book of Wisdom - John Doe ↵

# 3. Packages Used

Listed below are the imported python modules used in this project.

**Need more info?**

Click on any of the headings to read module documentation

## 3.1 Packages used in the user interface code, `frames.py`

tkinter: This package provides the building blocks of the user interface. It is the standard Python interface to the Tcl/Tk GUI toolkit.

tkinterdnd2: This package provides the functionality behind the 'drag & drop'.

## 3.2 Packages used in the data analysis code, `data_analysis.py`

xlrd: This package provides the tools required to read and analyse data from input excel files.

xlwt: This package provides the tools to generate the output excel file.