

# Avoiding

```
if __name__ == "__main__":
```

James Abel

Aug 18, 2019

[j@abel.co](mailto:j@abel.co)

```
def main():  
    print("Hello PyBay")
```

```
if __name__ == "__main__":  
    main()
```



```
def main():  
    print("Hello PyBay")
```

```
if __name__ == "__main__":  
    main()
```

```
if is_main():  
    main()
```



# is\_main.py

```
import inspect
```

```
def is_main():
```

```
    """
```

```
    Prettier version of __name__ == "__main__"
```

```
    :return: True if application is running as "main"
```

```
    """
```

```
    return inspect.stack()[1].frame.f_locals.get("__name__") == "__main__"
```

```
# pip install sundry
```

```
from sundry import is_main
```

```
def main():  
    print("Hello PyBay")
```

```
if is_main():  
    main()
```



# BACKUP

# The Zen of Python

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Flat is better than nested.
- Sparse is better than dense.
- Readability counts.
- Special cases aren't special enough to break the rules.
- Although practicality beats purity.
- Errors should never pass silently.
- Unless explicitly silenced.
- In the face of ambiguity, refuse the temptation to guess.
- There should be one-- and preferably only one --obvious way to do it.
- Although that way may not be obvious at first unless you're Dutch.
- Now is better than never.
- Although never is often better than *\*right\** now.
- If the implementation is hard to explain, it's a bad idea.
- If the implementation is easy to explain, it may be a good idea.
- Namespaces are one honking great idea -- let's do more of those!