

[View Source](#)

Homepage

class HomepagePref:

[View Source](#)

The HomepagePref Class pulls user preferences that have been set in MongoDB to then feed an object into the View.py file to ultimately be displayed on the homepage.

HomepagePref(username)

[View Source](#)

This is the constructor for the class. The constructor calls on method `getpref()` to begin. There are several global variables:

`self.user` - username associate with logged-in user

`self.preferences` - preferences that have been pulled from MongoDB

`self.timeframe` - Hourly or Daily aspect of preferences

`self.provider` - Weather Service Provider aspect of preferences

`self.datatype` - Type of weather data (observation, prior forecast, forecast) aspect of preferences

Args:

Username: username associated with the logged-in user

def converttime(self, time):

[View Source](#)

Converttime method is for converting the current time into the time an hour ago. The method also formats according to the standard format for other classes within the website.

Args:

time: the current time based on a `datetime.now()` filter

Returns:

Time an hour ago into a standard string format for the website purposes

def convertdateyesterday(self, date):

[View Source](#)

Convertdateyesterday method is for converting the date of "today" to "yesterday"'s date in a standard string format for the purpose of the website.

Args:

Date: "Today"'s date from the datetimeow() function

Returns:

Date: String format that is standard for the website purposes

```
def convertdateyesterday(self, date):
```

[View Source](#)

Convertdatetodaymethod is for converting the date of "today" into a standard string format for the purpose of the website.

Args:

Date: "Today"'s date from the datetimeow() function

Returns:

Date: String format that is standard for the website purposes

```
def getpref(self):
```

[View Source](#)

This method retrieves preference data based on user from MongoDB. This method calls displaypref() with arguments

```
def displaypref(self, datatype, ServiceProvider, HRorDAY):
```

[View Source](#)

This method parses through preferences to update global variables to be called from Views.py.

Args:

datatype: The type of data the user prefers (either weather observations, prior forecast, or forecast)

ServiceProvider: The service provider user selected to be applied to either prior forecast or forecast

HRorDay: User preferences of hourly or daily data. For Weather Observations, data returns yesterday or an hour prior observations

