[Skip to content](https://pythonguides.com/convert-html-page-to-pdf-using-django/#content)

Top of Form

Bottom of Form

* [Home](https://pythonguides.com/)
* [Blogs](https://pythonguides.com/blog/)
  + [Python Programming](https://pythonguides.com/python-programming/)
  + [Tkinter](https://pythonguides.com/tkinter/)
  + [Pandas](https://pythonguides.com/pandas/)
  + [NumPy](https://pythonguides.com/numpy/)
  + [Turtle](https://pythonguides.com/turtle/)
  + [Django](https://pythonguides.com/django/)
  + [Matplotlib](https://pythonguides.com/matplotlib/)
  + [TensorFlow](https://pythonguides.com/tensorflow/)
  + [PyTorch](https://pythonguides.com/pytorch/)
  + [Scikit-Learn](https://pythonguides.com/scikit-learn/)
  + [SciPy](https://pythonguides.com/scipy/)
* [ML](https://pythonguides.com/category/machine-learning/)
* [AI](https://pythonguides.com/category/ai/)
* [Training Course](https://pythonguides.com/python-and-machine-learning-training-course/)
* [YouTube](https://www.youtube.com/channel/UCNCMtijH5TFf4-P5rYQ6pRA?sub_confirmation=1)

Top of Form

Bottom of Form

**Convert HTML page to PDF using Django in Python + PDF Invoice**

November 19, 2021 by [Bijay Kumar](https://pythonguides.com/author/fewlines4biju/)

You might have faced various scenarios when you needed to convert an HTML document into a PDF file in Python using [Django](https://pythonguides.com/how-to-install-django/). In Django, you can implement this feature by using some libraries that we will discuss in this article.

In this article, you will learn to convert HTML pages to PDF using [Django](https://pythonguides.com/what-is-python-django/). You will see some examples so that you can understand the concept better.

* How to Convert HTML to PDF Django
* Django HTML to PDF example
* Django HTML to PDF with CSS
* Dynamically convert HTML to PDF in Django
* Django generate PDF invoice

Table of Contents

**How to Convert HTML to PDF Django**

You can use the [**xhtml2pdf** library](https://xhtml2pdf.readthedocs.io/en/latest/usage.html) in Django to convert an HTML document into a PDF file. Let us see how to use this library.

Firstly, you need to install the xhtml2pdf library using the **pip** command as shown below:

pip install xhtml2pdf

Once installed, you can use this library to convert an HTML document into a PDF file in Python. Let us discuss the further steps with the help of an example.

**Django HTML to PDF example**

Firstly, you have to create a new python file inside your [Django application](https://pythonguides.com/setup-django-project/). Suppose we have created with a name process.py

Inside this file, you will import the xhtml2pdf library and create a function that will convert an HTML document into a PDF file.

# importing the necessary libraries

from io import BytesIO

from django.http import HttpResponse

from django.template.loader import get\_template

from xhtml2pdf import pisa

# defining the function to convert an HTML file to a PDF file

def html\_to\_pdf(template\_src, context\_dict={}):

template = get\_template(template\_src)

html = template.render(context\_dict)

result = BytesIO()

pdf = pisa.pisaDocument(BytesIO(html.encode("ISO-8859-1")), result)

if not pdf.err:

return HttpResponse(result.getvalue(), content\_type='application/pdf')

return None

Now we will import this **process.py** file into our views.py file. In the views.py file, we will create a class-based view. In the view class, we will define a function that will invoke the function in the process.py file.

In the views.py file, we will also specify the HTML template that we want to be converted into a PDF file. The views.py file will look like this:

# importing the necessary libraries

from django.http import HttpResponse

from django.views.generic import View

from .process import html\_to\_pdf

#Creating a class based view

class GeneratePdf(View):

def get(self, request, \*args, \*\*kwargs):

# getting the template

pdf = html\_to\_pdf('result.html')

# rendering the template

return HttpResponse(pdf, content\_type='application/pdf')

For this demonstration purpose, I have created a sample HTML template that will be converted into a PDF file. The HTML template is:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Welcome to Python Guides</title>

</head>

<body>

<h2>This is a Sample HTML Content which is converted into a PDF file</h2><br>

<p>Hope you have understood how to convert an HTML document into a PDF file</p>

<p>You can also generate dynamic contenet using this technique.</p>

</body>

</html>

The last step is to define a URL endpoint where you want to view your PDF file in the url.py file. In my case, the urls.py file is:

from django.contrib import admin

from django.urls import path

#This will import our view that we have already created

from .views import GeneratePdf

urlpatterns = [

path('admin/', admin.site.urls),

path('pdf/', GeneratePdf.as\_view()),

]

In this urls.py file, I have defined a URL endpoint “pdf/”. When a request will be sent to this URL endpoint, the class-based view that we created in the views.py file will be called.

Now let us start the Django server and send a request to the /pdf/ endpoint.

Convert HTML page to PDF using Django

You can see that the content inside the HTML template is rendered as a PDF file. In this way, you can convert an HTML document into a PDF file.

Read [Get URL parameters in Django](https://pythonguides.com/get-url-parameters-in-django/)

**Django HTML to PDF with CSS**

In the above example, we simply converted an HTML page into a PDF file using Python Django. We can also apply some CSS to the HTML template and **convert this template into a PDF file in Python using Django**.

The approach will be almost the same as explained in the above section. In my case, the **process.py** file will be:

# importing the necessary libraries

from io import BytesIO

from django.http import HttpResponse

from django.template.loader import get\_template

from xhtml2pdf import pisa

# defining the function to convert an HTML file to a PDF file

def html\_to\_pdf(template\_src, context\_dict={}):

template = get\_template(template\_src)

html = template.render(context\_dict)

result = BytesIO()

pdf = pisa.pisaDocument(BytesIO(html.encode("ISO-8859-1")), result)

if not pdf.err:

return HttpResponse(result.getvalue(), content\_type='application/pdf')

return None

I will import this file into the **views.py** file. You can also write this code directly in the views.py file. But it is a good approach to write this code as a separate module.

Now the **views.py** file will look like this:

# importing the necessary libraries

from django.http import HttpResponse

from django.views.generic import View

from .process import html\_to\_pdf

#Creating a class based view

class GeneratePdf(View):

def get(self, request, \*args, \*\*kwargs):

# getting the template

pdf = html\_to\_pdf('result.html')

# rendering the template

return HttpResponse(pdf, content\_type='application/pdf')

I have created an HTML template with some random text. I have also used some CSS properties for this demonstration. This HTML template is saved with the name **results.html**:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Welcome to Python Guides</title>

</head>

<body>

<p style="color: black; font-size: xx-large; font-family:Arial, Helvetica, sans-serif, Geneva, Tahoma, sans-serif;"> Dear Customer,</p>

<p style="padding-left: 5em;">This is to inform you that your plan is expiring soon. Kindly, pay your monthly premium before expiration.</p>

<p style="font-size: x-large;">Thank You</p>

<p style="font-size: large; color: blue;"> Python Guides</p>

</body>

</html>

Now let us map the URL endpoint with a function created in the view.py file. This will be done in the urls.py file which is:

from django.contrib import admin

from django.urls import path

#This will import our view that we have already created

from .views import GeneratePdf

urlpatterns = [

path('admin/', admin.site.urls),

path('pdf/', GeneratePdf.as\_view()),

]

Save all the above files and start your Django server. Let us see the output:

Converted HTML to PDF with CSS

You can see that the content in the PDF file is written as it was in the HTML file i.e. our CSS reflected in the PDF file.

In this way, you can **convert an HTML file with CSS to a PDF file in Django**.

Also read, [How to get data from get request in Django](https://pythonguides.com/get-data-from-get-request-in-django/)

**Dynamically convert HTML to PDF in Django**

In real-life scenarios, you may need to display dynamic data to your users using a PDF file. For example, retrieving some data from the database and rendering it inside an HTML template, and converting it into a PDF file.

Some common use cases are generating documents like invoices for an e-commerce site, applications data, forms, resumes, etc.

In this section, I will demonstrate how you can retrieve data from your database and render it in your HTML template. Then I will convert this HTML template into a PDF document.

In Django, you have models that you can work with. I have created my **models.py** file as:

from django.db import models

class Employees(models.Model):

emp\_id = models.AutoField

first\_name = models.CharField(max\_length=20)

last\_name = models.CharField(max\_length=20)

department= models.CharField(max\_length=20)

The above model is equivalent to a table. I am using an SQLite database. Therefore I populated this table with some data using SQLite Studio. Now the table becomes:

| **id** | **first\_name** | **last\_name** | **department** |
| --- | --- | --- | --- |
| 1601 | Chris | Hemsworth | Research |
| 1602 | Helena | Williams | Finance |
| 1603 | Jerry | James | Marketing |
| 1604 | Dwayne | Morkel | Finance |

Employees Table

Now I will create a function to convert the HTML document into a PDF file in a separate python file, say **process.py.** You can also write it into the **views.py** file directly. But I will import it into the views.py file.

# importing the necessary libraries

from io import BytesIO

from django.http import HttpResponse

from django.template.loader import get\_template

from xhtml2pdf import pisa

# defining the function to convert an HTML file to a PDF file

def html\_to\_pdf(template\_src, context\_dict={}):

template = get\_template(template\_src)

html = template.render(context\_dict)

result = BytesIO()

pdf = pisa.pisaDocument(BytesIO(html.encode("ISO-8859-1")), result)

if not pdf.err:

return HttpResponse(result.getvalue(), content\_type='application/pdf')

return None

Now I will retrieve data from this table. This will be done in the **views.py** file. You can see that I have also imported the process.py file here:

# importing the necessary libraries

from django.http import HttpResponse

from django.views.generic import View

from app1 import models

from .process import html\_to\_pdf

from django.template.loader import render\_to\_string

#Creating a class based view

class GeneratePdf(View):

def get(self, request, \*args, \*\*kwargs):

data = models.Employees.objects.all().order\_by('first\_name')

open('templates/temp.html', "w").write(render\_to\_string('result.html', {'data': data}))

# Converting the HTML template into a PDF file

pdf = html\_to\_pdf('temp.html')

# rendering the template

return HttpResponse(pdf, content\_type='application/pdf')

In the above view, I retrieved data from the model. **Employees** are the table name in the database.

The **objects()** function is used to retrieve the records from the table and **all()** function specifies that all the records have to be retrieved.

Then the data is rendered in an HTML template and written in a temporary file.

The original HTML template i.e. **result.html** is:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Welcome to Python Guides</title>

<style>

h1 {

text-align: center;

}

td, th {

text-align: center;

}

</style>

</head>

<body>

<h1>List of Employees</h1>

<table>

<thead>

<tr>

<th>Employee ID</th>

<th>First Name</th>

<th>Last Name</th>

<th>Department</th>

</tr>

</thead>

<tbody>

{% for person in data %}

<tr>

<td>{{ person.id }}</td>

<td>{{ person.first\_name }}</td>

<td>{{ person.last\_name }}</td>

<td>{{ person.department }}</td>

</tr>

{% endfor %}

</tbody>

</table>

</body>

</html>

In the above template, we are using a loop to iterate over the records fetched from the database.

When this HTML template is rendered and written in a temporary file, it looks like this:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Welcome to Python Guides</title>

<style>

h1 {

text-align: center;

}

td, th {

text-align: center;

}

</style>

</head>

<body>

<h1>List of Employees</h1>

<table>

<thead>

<tr>

<th>Employee ID</th>

<th>First Name</th>

<th>Last Name</th>

<th>Department</th>

</tr>

</thead>

<tbody>

<tr>

<td>1601</td>

<td>Chris</td>

<td>Hemsworth</td>

<td>Research</td>

</tr>

<tr>

<td>1604</td>

<td>Dwayne</td>

<td>Morkel</td>

<td>Finance</td>

</tr>

<tr>

<td>1602</td>

<td>Helena</td>

<td>Williams</td>

<td>Finance</td>

</tr>

<tr>

<td>1603</td>

<td>Jerry</td>

<td>James</td>

<td>Marketing</td>

</tr>

</tbody>

</table>

</body>

</html>

After that, the new temporary HTML file is converted into a PDF file and the response is returned.

But we have to complete one more step i.e. define a path in the **urls.py** file and map it with our function in the view:

from django.contrib import admin

from django.urls import path

#This will import our view that we have already created

from .views import GeneratePdf

urlpatterns = [

path('admin/', admin.site.urls),

path('pdf/', GeneratePdf.as\_view()),

]

Now let us send a request to this endpoint in our browser.

How to Convert HTML to PDF Django

You can see in the above image that the records stored in the database are rendered inside a PDF document.

Hence, in this way, you can dynamically generate HTML content and convert it into a PDF file using the **xhtml2pdf** library.

Read [Python Django length filter](https://pythonguides.com/python-django-length-filter/)

**Django generate PDF invoice**

Generating a PDF invoice is one of the most common applications where we need to convert an HTML document into a PDF document using Django.

In this section, I will demonstrate how you can implement this functionality in your Django application.

* Firstly, I will make a function to convert an HTML file into a PDF file and import this into the **views.py** file of the project.
* Suppose I have created this function inside a file named **process.py**. The file will look like this:

# importing the necessary libraries

from io import BytesIO

from django.http import HttpResponse

from django.template.loader import get\_template

from xhtml2pdf import pisa

# defining the function to convert an HTML file to a PDF file

def html\_to\_pdf(template\_src, context\_dict={}):

template = get\_template(template\_src)

html = template.render(context\_dict)

result = BytesIO()

pdf = pisa.pisaDocument(BytesIO(html.encode("ISO-8859-1")), result)

if not pdf.err:

return HttpResponse(result.getvalue(), content\_type='application/pdf')

return None

* After this, I will import this file into the views.py file like:

# importing the necessary libraries

from django.http import HttpResponse

from django.views.generic import View

from app1 import models

from .process import html\_to\_pdf

from django.template.loader import render\_to\_string

#Creating a class based view

class GeneratePdf(View):

def get(self, request, \*args, \*\*kwargs):

data = models.invoices.objects.get(id= 1434)

open('templates/temp.html', "w").write(render\_to\_string('result.html',{'data': data} ))

# Converting the HTML template into a PDF file

pdf = html\_to\_pdf('temp.html')

# rendering the template

return HttpResponse(pdf, content\_type='application/pdf')

In the above code, you can see that I am loading the invoice data from a database using a Django model. Let me show you the **models.py** file where you can see the structure of my database:

from django.db import models

from django.db.models.base import Model

class invoices(models.Model):

name = models.CharField(max\_length= 20)

address = models.CharField(max\_length = 100)

pay\_method = models.CharField(max\_length = 15)

amount = models.FloatField(max\_length=10)

description = models.CharField(max\_length=50)

order\_date = models.DateField()

I have already populated this table with some sample records using SQLite studio, as I am using an SQLite database. The table looks like shown in the below image:

Invoices Table

Also, in the views.py file, you can see that I am passing the data to a Django template named **results.html**. The contents of this template are:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Untitled Document</title>

</head>

<body>

<table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td colspan="2"><img src="logo.png" width="150" /></td>

</tr>

<tr>

<td colspan="2">&nbsp;</td>

</tr>

<tr>

<td width="49%"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:600; font-size:15px;">Invoice Receipt</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px;">Invoice Number: {{data.id}}</td>

</tr>

<tr>

<td>&nbsp;</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:600; font-size:15px;">Service Provider </td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:400; font-size:14px;">Mahesh Traders</td>

</tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px;">1001A, The Capital B Wing, 10th Floor, Bandra Kurla Complex, Bandra (E), Mumbai, Contact: 7863087385 </td>

</tr>

<tr>

<td>&nbsp;</td>

</tr>

</table></td>

</tr>

</table></td>

<td width="51%" valign="top"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td align="right"><img src="logo.png" alt="" width="150" /></td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px;" align="right"></td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px;" align="right">&nbsp;</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px;" align="right">Order Date :{{data.order\_date}}</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:600; font-size:15px;" align="right">Customer Name</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px;" align="right">{{data.name}}</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px;" align="right">{{data.address}}</td>

</tr>

</table></td>

</tr>

<tr>

<td colspan="2">&nbsp;</td>

</tr>

<tr>

<td colspan="2"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:600; font-size:13px; border-top:1px solid #333; border-bottom:1px solid #333; border-left:1px solid #333; border-right:1px solid #333;" width="34%" height="32" align="center">Description</td>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:600; font-size:13px; border-top:1px solid #333; border-bottom:1px solid #333; border-right:1px solid #333;" width="30%" align="center">Bill Amount</td>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:600; font-size:13px; border-top:1px solid #333; border-bottom:1px solid #333; border-right:1px solid #333;" width="30%" align="center">Payment Method</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px; border-bottom:1px solid #333; border-left:1px solid #333; border-right:1px solid #333;" height="32" align="center">{{data.description}}</td>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px; border-bottom:1px solid #333; border-right:1px solid #333;" align="center">{{data.amount}}</td>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px; border-bottom:1px solid #333; border-right:1px solid #333;" align="center">{{data.pay\_method}}</td>

</tr>

</table></td>

</tr>

<tr>

<td colspan="2">&nbsp;</td>

</tr>

<tr>

<td colspan="2">&nbsp;</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:600; font-size:13px;" colspan="2">Please Note:</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px;" colspan="2">Dear Consumer, the bill payment will reflect in next 48 hours in your account. Please contact our customer support for any queries regarding this order.</td>

</tr>

<tr>

<td colspan="2">&nbsp;</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:600; font-size:13px;" colspan="2">Thank you for using our service</td>

</tr>

<td colspan="2">&nbsp;</td>

</tr>

<tr>

<td colspan="2">&nbsp;</td>

</tr>

<tr>

<td style="font-family:Verdana, Geneva, sans-serif; font-weight:300; font-size:13px;" colspan="2" align="center">(This is computer generated receipt and does not require physical signature.) </td>

</tr>

<tr>

<td colspan="2">&nbsp;</td>

</tr>

<tr>

<td colspan="2">&nbsp;</td>

</tr>

<tr>

<td colspan="2">&nbsp;</td>

</tr>

</table>

</body>

</html>

Now the final step is to map a URL endpoint with our code i.e. creating a path in the **urls.py** file. In my case, the urls.py file will be:

from django.contrib import admin

from django.urls import path

#This will import our view that we have already created

from .views import GeneratePdf

urlpatterns = [

path('admin/', admin.site.urls),

path('pdf/', GeneratePdf.as\_view()),

]

Once you have followed all the above steps, you can send a request to the URL endpoint that you defined in the urls.py file.

http://localhost:8000/pdf/

Generating a PDF invoice in Django

You can see that the values from the database are loaded in this PDF invoice. You can also customize your Django template by adding some styling using supported CSS( All CSS are not supported).

Thus, you might have got a basic idea of how you can generate a PDF invoice in Django.

Related Python Django tutorials:

* [Django for loop](https://pythonguides.com/django-for-loop/)
* [If statement in Django template](https://pythonguides.com/if-statement-in-django-template/)
* [Python Django get admin password](https://pythonguides.com/python-django-get-admin-password/)
* [Python Django vs Flask](https://pythonguides.com/python-django-vs-flask/)
* [Python Django group by](https://pythonguides.com/python-django-group-by/)
* [Python Django get](https://pythonguides.com/python-django-get/)
* [Contact Form with Django and SQLite](https://pythonguides.com/contact-form-with-django-and-sqlite/)

Hence, using the above ways, you can **convert an HTML document into a PDF document using Django in Python**.

* How to Convert HTML to PDF Django
* Django HTML to PDF example
* Django HTML to PDF with CSS
* Dynamically convert HTML to PDF in Django
* Django generate PDF invoice

[Bijay Kumar](https://pythonguides.com/author/fewlines4biju/)

Python is one of the most popular languages in the United States of America. I have been working with Python for a long time and I have expertise in working with various libraries on Tkinter, Pandas, NumPy, Turtle, Django, Matplotlib, Tensorflow, Scipy, Scikit-Learn, etc… I have experience in working with various clients in countries like United States, Canada, United Kingdom, Australia, New Zealand, etc. [Check out my profile](https://pythonguides.com/about-us/).

[enjoysharepoint.com/](https://enjoysharepoint.com/)

[How to get unique values in NumPy Array Python – Complete Tutorial](https://pythonguides.com/python-numpy-unique/)

[How to Draw Flower in Python Turtle](https://pythonguides.com/draw-flower-in-python-turtle/)

Top of Form

Search

Bottom of Form

**Follow us in Twitter & Facebook**

**Recent Posts**

* [How to add Google reCAPTCHA to Django Form](https://pythonguides.com/add-google-recaptcha-to-django-form/)
* [Django contact form with class based views](https://pythonguides.com/django-contact-form-with-class-based-views/)
* [How to Check if a Key Exists in a Python Dictionary](https://pythonguides.com/check-if-a-key-exists-in-a-python-dictionary/)
* [How to create a list of dictionary keys in python](https://pythonguides.com/create-a-list-of-dictionary-keys-in-python/)
* [How to Create a Dictionary from one list in Python](https://pythonguides.com/create-a-dictionary-from-one-list-in-python/)
* [About PythonGuides.com](https://pythonguides.com/about-us/)
* [Contact](https://pythonguides.com/contact/)
* [Privacy Policy](https://pythonguides.com/privacy-policy/)
* [Sitemap](https://pythonguides.com/sitemap/)

© 2022 PythonGuides.com

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  | |  |  | | --- | --- | |  |  | |  |