

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.

You will note that this code starts out in much the same way as our previous example. We still need to create an instance of PdfFileReader. But this time, we grab a page using the getPage method. PyPDF2 is zero-based, much like most things in Python, so when you pass it a one, it actually grabs the second page. The first page in this case is just an image, so it wouldn't have any text.

Interestingly, if you run this example you will find that it doesn't return any text. Instead all I got was a series of line break characters. Unfortunately, PyPDF2 has pretty limited support for extracting text. Even if it is able to extract text, it may not be in the order you expect and the spacing may be different as well.