

Generowanie plików PDF

Celem niniejszego artykułu nie będzie wcale wyjaśnienie tajemnic Adobe, a tylko pokazanie światu, jak uzyskać profesjonalnie wyglądające, generowane dynamicznie dokumenty PDF.

Na początek.

Kto z Was nigdy nie spotkał się sam na sam z PDFem? Co tu dużo mówić, chyba każdy zna ten format, mimo, że może nie wie o nim za wiele. Cóż, w zasadzie celem niniejszego arta nie będzie wcale wyjaśnienie tajemnic Adobe, a tylko pokazanie światu, jak zaskakująco łatwo uzyskać profesjonalnie wyglądające, generowane dynamicznie dokumenty PDF.

Po co to?

Coraz częściej ludzie poszukują jakiś sposobów na urozmaicenie strony albo przekazywanie danych w łatwy i szybki sposób. Co z tego, że poświęcisz mnóstwo pracy na tekst a przedstawisz go w sposób pozostawiający wiele do życzenia. Dno po prostu.

Co będzie potrzebne?

Serw, nawet darmowy, byle miał PHP, reszta jest nieważna

+

darmowy komponent FPDF, który znajdziesz [tutaj](#), a na którym opiera się cały ten art.

+

konsolowy program ttf2pt1.exe, który w zasadzie krąży po sieci, ale ja znalazłem go [>tutaj](#).

+

chwila wolnego czasu, troche cierpliwości.

No to start.

- 1) Na początek ściągasz komponent FPDF i rozpakowujesz go. Bierzesz sobie plik `fpdf.php` oraz cały katalog `font` i kopiujesz na serwa.
- 2) Każdy dokument PDF posiada kodowanie, ale próżno szukać tam `iso-8859-2`, albo `utf-8`. Żeby mieć polskie znaki, musisz dokonać pewnych modyfikacji. Zakładam, że chcesz mieć w dokumencie polskie znaki, jeżeli jest inaczej, możesz przejść od razu do punktu 8).
- 3) Bierzesz plik czcionki, którą chcesz mieć w swoim dokumencie, ja np. wybrałem `arial.ttf`. Ściągasz [ttf2pt1.exe](#). Kopiujesz plik czcionki, oraz plik `iso-8859-2.map` z katalogu `font` komponentu FPDF do katalogu z programem `ttf2pt1.exe`. Teraz musisz uruchomić konsolę i wpisać w niej poniższą linijkę:
`ttf2pt1 -b -L iso-8859-2.map arial.ttf arialpl`

Jeżeli operacja przebiegła poprawnie (przeleciało przed oczyma parę ekranów), a na końcu masz coś takiego:

```
Finished - font files created
Creating file arialpl.pfb
Converting file arialpl.t1a
Removing file arialpl.t1a
```

znaczy, że konwersja czcionki przebiegła pomyślnie, ale to jeszcze nie koniec.

- 4) Do katalogu `/font/makefont` komponentu FPDF kopiujesz powstałe pliki `.afm` i `.pfb`, dla mnie były to

arialpl.afm i arialpl.pfb.

5) Teraz tworzysz skrypt, w którym wpisujesz:

```
<?php
require('makefont.php');
//znowu konwertujesz czcionkę
Makefont('arialpl.pfb', 'arialpl.afm', 'iso-8859-2');
?>
```

6) Zapisujesz plik, np jako convert.php w aktualnym katalogu, po czym wywołujesz go sobie w przeglądarce. Powinieneś dostać komunikat o pomyślnej konwersji (*Font definition file generated*), a w katalogu ze skryptem (przypominam /font/makefont nie inny) powinieneś zauważyć dwa nowe pliki - arialpl.php i arialpl.z. Twoja czcionka jest już gotowa do użycia, jeśli masz już wszystkie potrzebne czcionki, przejdź do punktu następnego, jeśli nie, wróć do punktu 3).

7) Wszystkie pliki nazwa_czcionki.php i nazwa_czcionki.z kopiujesz folder wyżej (/font). Teraz już możesz użyć swoich czcionek w dokumencie. Jak pewnie zauważyłeś, komponent ma też parę czcionek wbudowanych, są to m.in. Helvetica, Symbol, Times itp.

Czas na PDFa

8) Nie zostaje już nic, jak nareszcie wziąć się za skrypt generujący PDFy. To na początek:

```
<?php
define('FPDF_FONTPATH','font/'); //definiuje katalog z czcionkami komponentu
require('fpdf.php'); //odniesienie do skryptu komponentu
$pdf=new FPDF();
```

Komponent FPDF jest klasą, dzięki czemu dokument tworzy się bardzo łatwo, jedyne co może sprawiać problemy, to orientacja przestrzenna tekstu i obiektów, ale i na to znajdzie się sposób.

Teraz pora na nowy dokument, dopisujesz:

```
$pdf->Open(); //otwiera nowy dokument
$pdf->AddPage(); //dodaje nową stronę do dokumentu
```

Żeby zapełnić dokument tekstem, najpierw należy ustawić czcionkę.

W ten sposób, jeśli stworzyłeś swój plik czcionki:

```
$pdf->AddFont('arialpl','', 'arialpl.php'); //dodaje swoją czcionkę arialpl do dokumentu
$pdf->SetFont('arialpl','', 12); //ustawia czcionkę arialpl, rozmiar 12
```

Albo w ten, jeśli będziesz używał czcionki wbudowanej:

```
$pdf->SetFont('Times','', 12); //ustawia wbudowaną czcionkę Times, rozmiar 12
```

AddFont(string rodzina [, string styl [, string plik]])

SetFont(string rodzina [, string styl [, float rozmiar]])

rodzina - nazwa czcionki, zazwyczaj taka sama, jak nazwa pliku, tylko bez rozszerzenia.

styl - pusty tekst: zwykły (domyślnie) 'B': wytłuszczona 'I': kursywa 'BI' lub 'IB': wytłuszczona kursywa.

plik - nazwa pliku *.php czcionki.

rozmiar - rozmiar czcionki w pikselach.

Później, żeby zmienić tylko rozmiar lub kolor czcionki, wystarczy instrukcja:

```
$pdf->SetFontSize(10); //zmienia rozmiar ostatnio ustawionej czcionki na 10  
$pdf->SetTextColor(,255); //zmienia kolor czcionki na RGB(0,0,255)
```

Funkcji \$pdf->AddFont() używaj za każdym razem, gdy chcesz mieć jakąś nową, niestandardową czcionkę w dokumencie, a \$pdf->SetFont(), żeby zmienić czcionkę, którą piszesz. Zauważ, że jeśli chcesz użyć czcionki wbudowanej w komponent nie musisz jej dodawać.

Jeżeli w dokumencie ani razu nie wywołasz funkcji \$pdf->SetFont(), będziesz pisał domyślnym dla komponentu zestawem znaków (bodajże Helvetica).

9) Teraz pora, żebyś umieścił w dokumencie swój pierwszy tekst. Żeby to zrobić, masz do wyboru kilka funkcji, każda działa nieco inaczej.

Cell(float w [, float h [, string txt [, mixed ramka [, int ln
[, string wyrównanie [, int tło [, mixed link]]]]]])

- **wpisuje ramkę tekstową.**

MultiCell(float w, float h, string txt [, mixed ramka
[, string wyrównanie [, int tło]])

- **wpisuje wieloliniowy tekst.**

Text(float x, float y, string txt)

- **wpisuje ciąg znaków.**

Write(float h, string txt [, mixed link])

- **wlewa tekst.**

gdzie:

w - szerokość pola. Jeśli 0, pole rozszerza się aż do prawego marginesu.

h - wysokość pola. Domyślnie: 0. (W Multicell - wysokość linijki!)

txt - tekst do umieszczenia w polu. Domyślnie: pusty tekst.

ramka - wskazuje czy zostanie narysowana ramka. Wartość musi być liczbą:

0: bez ramki

1: z ramką

albo tekstem zawierającym niektóre lub wszystkie z następujących znaków (w dowolnej kolejności):

L: lewa

T: górna

R: prawa

B: dolna

Domyślnie: 0.

ln - wskazuje dokąd przeniesie się kursor po wywołaniu metody. Możliwe wartości to:

0: w prawo

1: na początek następnej linii

2: poniżej

Ustawienie 1 jest równoważne użyciu 0 a następnie wywołaniu Ln(). Domyślnie: 0.

wyrównanie - pozwala tekst wyśrodkować albo wyrównać do marginesu. Możliwe wartości:

L albo pusty tekst: do lewej (wartość domyślna)

C: wyśrodkowanie

R: do prawej

J: justowanie (tylko w Multicell)

tlo - wskazuje czy tło pola ma być przezroczyste (0) czy wypełnione bieżącym kolorem (1). Domyślnie: 0.

link - URL albo identyfikator zwrócony przez AddLink().

```
$pdf->SetFont('arialpl','',20);
$pdf->Text(10,10, 'Witaj świecie. To jest tekst bez zawijania'); //tekst bez zawijania na pozycji x=10, y=10
$pdf->SetFont('arialpl','',14);
$pdf->Multicell(4, 'Ten tekst z zamierzenia miał być długi, w każdym razie raczej nie powinien zmieścić się w
jednej linijce, ale nie ma żadnego problemu, funkcja Multicell() służy do wprowadzania tekstu z zawijaniem, ba
jeśli tekst będzie dłuższy od strony, utworzy ona nową! ', 'J'); //tekst wieloliniowy o szerokości do prawej linii,
wysokości linii 4, bez ramki, wyjustowany, bez tła
/* Dopisuje niebieski podkreślony odnośnik */
$pdf->SetFont('arialpl','',14);
$pdf->Write(10,'Zapraszam na ');
$pdf->SetTextColor(.,255); //zmienia kolor czcionki
$pdf->SetFont('','U'); //zmienia styl czcionki na podkreślenie
$pdf->Write(10,'4programmers.net','http://4programmers.net');
```

Gdzie jest kursor?

10) Jak już mówiłem, dość sporym problemem w pisaniu PDFów jest to, że musisz ręcznie ustawiać pozycję wszystkich tekstów i elementów. Czasami trudno się połąpać, często zdarza się, że jakieś teksty zachodzą na siebie. A do tego, przecież masz tworzyć dokumenty dynamicznie, gdzie długość tekstu nie będzie za każdym razem taka sama. Żeby jakoś to wszystko zorganizować, wykorzystasz 6 prostych funkcji:

GetX()

GetY

SetXY(float x,y)

SetX(float x)

SetY(float y)

a także Ln(float h)

Pierwsze dwie służą do zwrócenia współrzędnej x i y bieżącej pozycji kursora, kolejne 3 do ustawienia pozycji kursora, a ostatnia służy do łamania linii (X bieżącej pozycji kursora wraca do lewego marginesu, Y jest zwiększana o wielkość parametru h).

Po każdym wywołaniu funkcji wstawiającej do dokumentu tekst lub inny element, kursor jest przesuwany na dół, dzięki czemu nie musisz obliczać wysokości tekstu i na tej podstawie przesuwać kursor.

11) Teraz pod tekst wstawisz poziomą linię:

```
$iks = $pdf->GetX;
$igrek = $pdf->GetY;
$pdf->Line($iks, $igrek+2,200, $igrek+2); //wstawia linię 2mm pod tekstem, o długości 200mm.
```

Line(float x1, float y1, float x2, float y2) // **linia**

Rect(float x, float y, float w, float h [, string style]) // **prostokąt**

x1 - współrzędna X początku linii (lewego górnego rogu prostokąta).

y1 - współrzędna Y początku linii (lewego górnego rogu prostokąta)..

x2 - współrzędna X końca linii.

y2 - współrzędna Y końca linii.

w - szerokość prostokąta

h - wysokość prostokąta

style - sposób rysowania. Możliwe wartości to:

D albo pusty tekst: tylko kontur (wartość domyślna)

F: tylko wypełnienie

DF or FD: wypełnienie i kontur

12) Żeby linie, prostokąty i ramki nie były czarne, musisz użyć następujących funkcji do zmiany koloru linii i wypełnienia:

```
SetDrawColor(int r [, int g, int b]) // ustawia kolor rysowania na RGB(r,g,b)
```

```
SetFillColor(int r [, int g, int b]) // ustawia kolor wypełnienia na RGB(r,g,b)
```

```
/* narysuje granatowy prostokąt z zielonym wypełnieniem */
```

```
$pdf->SetDrawColor(170,255,64);
```

```
$pdf->SetFillColor(54,255,102);
```

```
$pdf->Rect($iks+20, $igrek+20,200,100);
```

13) Na koniec, do naszego dokumentu wstawisz grafikę. Może to być plik JPEG, lub PNG o dowolnym rozmiarze i rozdzielczości [tak twierdzi autor komponentu], jednak nie może być progresywny, ani mieć przeplotu (Interlacing), a także nie może posiadać kanału alpha.

```
Image(string plik, float x, float y, float w [, float h [, string typ [, mixed link]]])
```

plik - nazwa pliku z obrazkiem.

x - współrzędna X lewego górnego rogu obrazka.

y - współrzędna Y lewego górnego rogu obrazka.

w - szerokość obrazka na stronie. Jeśli równa jest zero, obliczana jest proporcjonalnie do wysokości oryginalnego obrazka.

h - wysokość obrazka na stronie. Jeśli nie podana lub równa zero, obliczana automatycznie, proporcjonalnie do szerokości.

typ - format pliku obrazka. Przyjmuje wartości (wielkość liter bez znaczenia): JPG, JPEG, PNG. Jeśli nie jest podana, ustalana jest na podstawie rozszerzenia nazwy pliku.

link - URL albo identyfikator zwracany przez AddLink().

14) Obrazek wstawisz na następnej stronie, ale najpierw zrobisz odnośnik do tej strony na pierwszej stronie. Do tego wykorzystasz kilka funkcji:

```
int AddLink()
```

```
SetLink(int link [, float y [, int strona]])
```

```
Link(float x, float y, float w, float h, mixed link)
```

gdzie:

link - unikalny dla każdego linku identyfikator stworzony przez funkcję AddLink() lub w wypadku funkcji Link() także URL

y - współrzędna oznaczająca, do której linijki na prowadzić link, 0 oznacza górę strony

strona - numer strony docelowej, -1 oznacza stronę bieżącą.

w - szerokość linku

h - wysokość linku

Wcześniej, żeby utworzyć proste hiperłącze, użyłeś tekstowej funkcji Write(), a cel odnośnika wpisałeś jako URL.

Żeby zrobić wewnętrzny link, najpierw musisz stworzyć identyfikator dla tego linka:

```
$ident = $pdf->AddLink();  
$pdf->SetLink($ident,,2); //tworzy (ale nie wstawia do dokumentu!) link do strony 2
```

Link nie jest tekstem, ani obrazem, tylko niewidocznym, aktywnym prostokątem, który możesz podłożyć pod dowolny element na stronie. Umieśćisz teraz swój link pod tekstem:

```
$tekst = 'Tu znajduje się link do następnej strony!';  
$dlugosc_tekstu = $pdf->GetStringWidth($tekst); //oblicza długość tekstu  
$pdf->Text($pdf->GetX(),$pdf->GetY(),$tekst); //wstawia tekst do dokumentu  
$pdf->Link($pdf->GetX(),$pdf->GetY(),$dlugosc_tekstu,20, $ident); //wstawia pod tekstem link do dokumentu
```

Pamiętaj: Odnośniki tekstowe i graficzne są zwykle wstawiane za pomocą Cell(), Write() albo Image(), nie ma potrzeby tworzenia linków pod nimi w podany wyżej sposób, chyba, że chcesz żeby np. tylko część obrazka była aktywna itp.

15) No, a teraz czas a obrazek.

```
$pdf->AddPage(); //dodaje nową stronę.  
$pdf->Image('C:\obraz.png', $pdf->GetX()+10, $pdf->GetY()+10, 123, 240, 'PNG');  
$pdf->SetFont('arialpl','8');  
$pdf->SetTextColor(,,);  
$pdf->Text($pdf->GetX(),$pdf->GetY()+1, 'i to by było na tyle');
```

Co jeszcze można zrobić?

PDF jest świetnym formatem o tyle, że umożliwia zabezpieczenie naszych danych przed kopiowaniem, czy też edycją, a także posiada możliwość kompresji. Przy pomocy darmowego komponentu można wstawić dane autorskie i przede wszystkim, co bardzo się przyda - można skompresować nieco (nieraz nawet o ponad połowę) nasz dokument.

16) O to co trzeba zrobić, żeby skompresować dokument:

```
$pdf->SetCompression(true); //włącza kompresję dokumentu
```

17) Jeśli nie interesuje Cię, jak ustawić inne właściwości dokumentu, takie jak nazwa autora, prawa autorskie, a także szerokość marginesów, możesz od razu przejść do punktu 18).

```
$pdf->SetAuthor('Ceer'); //ustawia autora dokumentu  
$pdf->SetCreator('Dokument generowany przy pomocy skryptu'); //ustawia generator dokumentu  
$pdf->SetKeywords('słowo_kluczowe1, słowo_kluczowe2'); //ustawia słowa kluczowe dokumentu  
$pdf->SetSubject('Nauka dynamicznego tworzenia PDFów'); //ustawia temat dokumentu  
$pdf->SetTitle('Jak łatwo stworzyć PDFa'); //ustawia tytuł dokumentu
```

```
$pdf->SetDisplayMode(100); //domyślne powiększenie dokumentu w przeglądarce
```

/*

SetDisplayMode(mixed zoom [, string layout])

zoom - powiększenie. Przyjmuje jedną z następujących wartości:

fullpage: cała strona mieści się w oknie

fullwidth: szerokość strony wypełnia całe okno

real: wielkość rzeczywista (równoważne powiększeniu 100%)
default: domyślny
albo liczba określająca skalę powiększenia
layout - układ strony. Wartości:
single: pojedyncze strony
continuous: kolejne strony
two: po dwie obok siebie
default: domyślny przeglądarki
Wartość domyślna to continuous.

I jeszcze na koniec:

```
$pdf->SetMargins(float lewy, float górny [, float prawy])  
ustawia marginesy dla dokumentu  
*/
```

18) Kiedy już wypełniłeś swój dokument czym tylko chciałeś, pora go wygenerować. Robi się to w następujący sposób:

```
$pdf->Output(); //zamyka i generuje dokument  
?>
```

Po tym zapisie i po uruchomieniu skryptu, powinieneś ujrzeć w przeglądarce zapytanie, czy chcesz otworzyć, czy zapisać plik PDF na dysk. Funkcja Output() bez parametru wysyła plik do nagłówka przeglądarki, a jeśli jako parametr podasz nazwę pliku, dokument zostanie zapisany na serwerze.

Cały skrypt powinien wyglądać mniej więcej tak:

```
<?php  
define('FPDF_FONTPATH','font'); //definiuje katalog z czcionkami komponentu  
require('fpdf.php'); //odniesienie do skryptu komponentu  
$pdf=new FPDF();  
$pdf->Open(); //otwiera nowy dokument  
$pdf->AddPage(); //dodaje nową stronę do dokumentu  
$pdf->AddFont('arialpl','arialpl.php'); //dodaje Twoją czcionkę arialpl do dokumentu  
$pdf->SetFont('arialpl','',20); //ustawia czcionkę arialpl, rozmiar 20  
$pdf->Text(10,10, 'Witaj świecie. To jest tekst bez zawijania'); //tekst bez zawijania na pozycji x=10, y=10  
$pdf->SetFont('arialpl','',14);  
$pdf->Multicell(4, 'Ten tekst z zamierzenia miał być długi, w każdym razie raczej nie powinien zmieścić się w jednej linijce, ale nie ma żadnego problemu, funkcja Multicell() służy do wprowadzania tekstu z zawijaniem, ba jeśli tekst będzie dłuższy od strony, utworzy ona nową! ', 'J'); //tekst wieloliniowy o szerokości do prawej linii, wysokości linii 4, bez ramki, wyjustowany, bez tła  
/* Dopisuje niebieski podkreślony odnośnik */  
$pdf->SetFont('arialpl','',14);  
$pdf->Write(10,'Zapraszam na ');  
$pdf->SetTextColor(,255); //zmienia kolor czcionki  
$pdf->SetFont('','U'); //zmienia styl czcionki na podkreślenie  
$pdf->Write(10,'4programmers.net','http://4programmers.net!');  
$iks = $pdf->GetX;  
$igrek = $pdf->GetY;  
$pdf->Line($iks, $igrek+2,200, $igrek+2); //wstawia linię 2mm pod tekstem, o długości 200mm.  
/* narysuje granatowy prostokąt z zielonym wypełnieniem */  
$pdf->SetDrawColor(170,255,64);  
$pdf->SetFillColor(54,255,102);  
$pdf->Rect($iks+20, $igrek+20,200,100);  
$ident = $pdf->AddLink();  
$pdf->SetLink($ident,2); //tworzy (ale nie wstawia do dokumentu!) link do strony 2  
$tekst = 'Tu znajduje się link do następnej strony!';
```

```
$dlugosc_tekstu = $pdf->GetStringWidth($tekst); //oblicza długość tekstu
$pdf->Text($pdf->GetX(),$pdf->GetY(),$tekst); //wstawia tekst do dokumentu
$pdf->Link($pdf->GetX(),$pdf->GetY(),$dlugosc_tekstu,20, $ident); //wstawia pod tekstem link do dokumentu
$pdf->AddPage(); //dodaje nową stronę.
$pdf->Image('C:\obraz.png', $pdf->GetX()+10, $pdf->GetY()+10, 123, 240, 'PNG');
$pdf->SetFont('arialpl','',8);
$pdf->SetTextColor(,);
$pdf->Text($pdf->GetX(),$pdf->GetY()+1, 'i to by było na tyle');
$pdf->SetCompression(true); //włącza kompresję dokumentu

/* a poniższe tylko dla ambitnych */
$pdf->SetAuthor('Ceer'); //ustawia autora dokumentu
$pdf->SetCreator('Dokument generowany przy pomocy skryptu'); //ustawia generator dokumentu
$pdf->SetKeywords('słowo_kluczowe1, słowo_kluczowe2'); //ustawia słowa kluczowe dokumentu
$pdf->SetSubject('Nauka dynamicznego tworzenia PDFów'); //ustawia temat dokumentu
$pdf->SetTitle('Jak łatwo stworzyć PDFa'); //ustawia tytuł dokumentu

$pdf->SetDisplayMode(100); //domyślne powiększenie dokumentu w przeglądarce
$pdf->SetMargins(20, 20, 20); //ustawia marginesy dla dokumentu

/* kończy zabawę i generuje dokument */
$pdf->Output(); //zamyka i generuje dokument
?>
```

To byłoby na tyle, zachęcam do wczytania się do polskiej pomocy na stronie komponenty FPDF, a także wykonania paru tutoriali. Warto pamiętać, że PDF jest ciągle rozwijany, a na stronie komponentu wciąż przybywa dodatków, jest nawet taki, który konwertuje HTMLa do PDFa, warto mu się przyjrzeć.

Źródło: 4programmers.net. Treść udostępniona na zasadach licencji [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/)

Autor: 4programmers.net

Artykuł pobrano ze strony eioba.pl