

Technologie informacyjne

przed 8 wykładem

Andrzej Giniewicz

17.04.2024

Ostatnia partia materiału dotyczącego \LaTeX a podzielona jest na dwa bloki. Pierwszy dotyczy umieszczania przypisów i cytowań, drugi dotyczy materiałów takich jak ilustracje, tabele oraz kody programów. Materiał omówiony podczas trzech wykładów nie jest wyczerpujący, dlatego na zakończenie notatek przedstawione zostaną dodatkowe źródła, z których warto skorzystać.

1 Przypisy

Przypis stanowi formę uzupełnienia treści. Może być umieszczony na dole strony, na marginesie, na końcu rozdziału lub na końcu tekstu głównego w osobnej sekcji. Charakterystycznym elementem każdego przypisu jest to, że w miejscu jego wystąpienia pojawia się odsyłacz. Odsyłacz to numer pozwalający zlokalizować, która informacja pomocnicza i gdzie umieszczona odnosi się do tekstu głównego. Najczęściej rozróżniamy cztery rodzaje przypisów:

1. przypisy rzeczowe, które zawierają treść uzupełniającą lub komentującą tekst główny,
2. przypisy słownikowe, które objaśniają termin lub interpretację słów użytych w tekście głównym,
3. przypisy informacyjne, które zawierają dodatkowe ułatwienia w korzystaniu z tekstu głównego,
4. przypisy bibliograficzne, nazywane też cytowaniami, które podają informacje bibliograficzne¹ o źródłach, na podstawie których opracowano tekst główny.

W polskiej tradycji typograficznej, przypisy najczęściej umieszcza się na dole strony lub w osobnej sekcji na końcu tekstu głównego. Pierwsza forma stosowana jest do książek o charakterze praktycznym oraz edukacyjnym, druga do książek o charakterze naukowym. Ogólna zasada mówi, że jeśli przypisy mogą pojawiać się wielokrotnie, powinny być

¹Informacje bibliograficzne, czyli takie, które pozwalają jednoznacznie zidentyfikować źródło — zastanów się, jakim rodzajem przypisu jest ta notka u dołu strony?

umieszczone w osobnej sekcji, ponieważ pozwala to uniknąć duplikacji treści. W pracach naukowych często wielokrotnie powołujemy się na te same publikacje w różnych miejscach treści, zatem jest to naturalne, że bibliografie w pracach naukowych umieszczone są na końcu.

W \LaTeX u domyślnie mamy komendy `\footnote{przypis}` oraz `\cite{etykieta}` do umieszczania przypisów u dołu strony oraz przypisów bibliograficznych na końcu publikacji. Jest też komenda `\marginpar{przypis}` umieszczająca przypis na marginesie, jest jednak domyślnie wyłączona w klasach `mwbk`, `mwrep` oraz `mwart`, ponieważ w polskiej tradycji typograficznej marginalia są rzadko spotykane. Wyłączenie ich pozwoliło autorowi klas na zaoszczędzenie miejsca na marginesie. Niemniej jednak można je włączyć, podając opcję `withmarginpar` podczas ładowania klasy. Trzeba pamiętać, że zwiększy to marginesy zewnętrzne w stosunku do standardowego układu kolumny.

```
\documentclass[withmarginpar]{mwrep}
```

Do umieszczania przypisów innych niż bibliograficzne na końcu rozdziału lub na końcu tekstu głównego służy pakiet `endnotes`. Do umieszczania cytowań na końcu rozdziału zamiast na końcu tekstu głównego, służy pakiet `chapterbib`. Istnieje więcej różnych metod sterowania przypisami, jednak do większości zastosowań naukowych, w zupełności wystarczą podstawowe komendy `\cite` oraz `\footnote`.

Przypisy u dołu strony najczęściej oddzielone są od kolumny linią poziomą. Numeracja przypisów powinna być umieszczona przed interpunkcją, bezpośrednio przylegając do słowa. Sama treść stopki powinna kończyć się odpowiednim znakiem interpunkcyjnym. Przykład fragmentu przypisu znajduje się poniżej.

```
uzupełniany tekst\footnote{Treść przypisu.}. Dalszy ciąg tekstu
```

Zwróćmy uwagę, że kropka jest przed klamrą zamykającą i tuż po niej, przy czym ta pierwsza kończy zdanie będące treścią przypisu, natomiast ta druga kończy zdanie będące uzupełniającym tekstem.

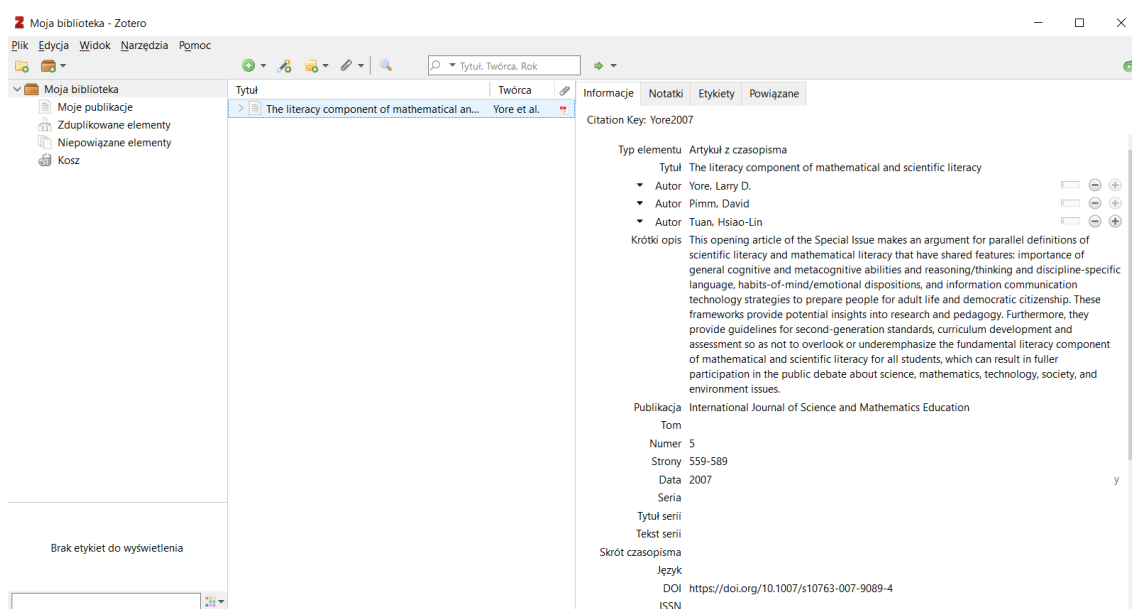
Zarządzanie przypisami bibliograficznymi, czyli cytowaniami, jest nieco trudniejsze. \LaTeX potrzebuje kilku rzeczy:

1. bazy danych cytowań bibliograficznych w pliku `.bib`,
2. pakietu zbierającego i formatującego cytowania,
3. narzędzia do sortowania użytych cytowań.

Podczas pierwszej kompilacji \LaTeX zbiera informacje, jakich cytowań używamy, następnie narzędzie do przetwarzania bibliografii wybiera z bazy cytowań odpowiednie wpisy i sortuje je. Po kolejnej kompilacji, w miejscu użycia odpowiedniej komendy, zależnej od wybranego pakietu do cytowań, pojawi się bibliografia, najczęściej w postaci osobnego rozdziału.

Zacznijmy od opisu programu do zarządzania bazą cytowań bibliograficznych. Podczas zajęć korzystamy z programu Zotero² oraz dodatku Better BibTeX³. Opis instalacji znajduje się na stronie wykładu. Po uruchomieniu programu po raz pierwszy wchodzimy do ustawień poprzez menu „Edycja” i „Preferencje”, wchodzimy do zakładki „Better BibTeX” i ustawiamy pole „Citation key” na wartość `auth + year`. Jeśli nie widzisz zakładki, oznacza to, że wtyczka „Better BibTeX” jest nieprawidłowo zainstalowana.

Dodawanie nowych wpisów polega na wybieraniu opcji „Nowy element” z menu „Plik” lub kliknięciu zielonego przycisku z symbolem *plus*. Następnie w interfejsie wybieramy typ publikacji, który cytujemy. Może być to artykuł, książka, rozdział książki, podręcznik, strona internetowa, dokumentacja, czyjaś praca dyplomowa, oprogramowanie, zbiór danych i wiele innych. W zależności od wybranego typu, cytowanego źródła, różnie będzie wyglądał dalszy interfejs. Założmy, że dodajemy artykuł. Wypełniamy pola, które znajdujemy w źródle. Przykładowy artykuł z uzupełnionymi polami znajduje się na poniższym zrzucie ekranu.



Zwróćmy uwagę na kilka rzeczy. Ważnym elementem jest klucz również widoczny na pierwszej stronie formularza. W tym przypadku jest to automatycznie generowany `Lore2007` składający się z nazwiska pierwszego autora i roku publikacji artykułu. W tekście głównym, który piszemy, aby zacytować tę pozycję, użyjemy `\cite{Lore2007}`. Przykładowy artykuł na zdjęciu ma również inne opcjonalne pola. Między innymi wypełnione zostało pole z krótkim streszczeniem, które pozwala wyświetlić streszczenie artykułu podczas przeglądania pozycji na liście. Dodany został również artykuł w formacie PDF, aby kliknięciem na pozycji w spisie można było otworzyć jego treść. Uzupełniony został również DOI dokumentu⁴, dzięki czemu jedno kliknięcie prowadzi na stronę wydawcy artykułu, gdzie można znaleźć dodatkowe informacje.

²Program Zotero jest dostępny za darmo na stronie <https://www.zotero.org/download/>.

³Dodatek jest dostępny na stronie <https://retorque.re/zotero-better-bibtex/>.

⁴W przypadku cytowanego artykułu DOI to <https://doi.org/10.1007/s10763-007-9089-4>. DOI to uniwersalne numery dokumentów. Więcej na ich temat możesz przeczytać na stronie <https://www.doi.org/>.

Po dodaniu wszystkich informacji w Zotero klikamy „Plik” oraz „Eksportuj bibliotekę”. Z listy rozwijanej wybieramy „Better Bib \LaTeX ” i podajemy nazwę pliku bibliografia.bib, który zapisujemy w tym samym katalogu co nasz plik .tex. Najważniejsze elementy wpisu wyszczególnione zostały w przykładzie poniżej. Ze względu na miejsce, z przykładu usunięte zostały informacje pomocnicze, takie jak streszczenie oraz plik PDF. Pliki .bib mają określoną składnię, nie musimy ich jednak edytować ręcznie. Jeśli chcielibyśmy, możemy to zrobić w dowolnym edytorze tekstu, na przykład Visual Studio Code.

```
@article{Yore2007,  
  title = {The Literacy Component of Mathematical and Scientific Literacy},  
  author = {Yore, Larry D. and Pimm, David and Tuan, Hsiao-Lin},  
  date = {2007},  
  journaltitle = {International Journal of Science and Mathematics Education},  
  number = {5},  
  pages = {559--589},  
  doi = {10.1007/s10763-007-9089-4}  
}
```

Zaletą korzystania z programu takiego jak Zotero jest to, że wszystkie źródła do pisania pracy mamy w jednym miejscu. Możemy w bazie danych mieć więcej artykułów, niż rzeczywiście użyjemy w pracy. Gdy natrafimy na ciekawy artykuł, warto dodać go do bazy. Dzięki grupom i niestandardowym polom możemy przykładowo osobno pokolorować artykuły na liście. Często praktyką jest stworzenie grup „do przeczytania”, „do zacytowania”, „odrzucone” oraz „zacytowane”. Zotero posiada również dodatek do przeglądarki pozwalający dodać materiały znalezione w Internecie do biblioteki wpisów, co znacząco przyspiesza proces jej tworzenia. Zachęcam do zapoznania się z dokumentacją programu Zotero⁵.

Po przygotowaniu bibliografii musimy odpowiednio skonfigurować dokument, dodając do preambuły informacje o stylu oraz bazie danych cytowań. Do preambuły dopisujemy linie

```
\usepackage[backend=biber,style=numeric-comp,sortlocale=pl_PL]{biblatex}  
\addbibresource{bibliografia.bib}
```

Pakiet Bib \LaTeX jest nowoczesnym pakietem do zarządzania bibliografią, który dobrze obsługuje kodowanie UTF-8 oraz sortowanie w różnych językach. Przekazujemy mu kilka opcji — backend informuje \LaTeX a, jakim narzędziem będzie sortowana bibliografia. Biber jest odpowiednikiem programu Bib \TeX , który podobnie jak Bib \TeX poprawnie obsługuje kodowanie UTF-8. Styl cytowań numeric-comp odpowiada za tak zwane cytowania w systemie vancouverkim. Jest to popularny system w pracach z nauk przyrodniczych i matematycznych. Często stosowany jest również system harwardzki⁶. Dzięki opcji sortlocale, cytowania zostaną posortowane zgodnie z polskimi zasadami. Linia z komendą addbibresource pozwala podać nazwę pliku, w którym szukane będą cytowania.

Na końcu dokumentu, po wszystkich rozdziałach, ale przed `\end{document}`, należy dopisać

⁵Proponuję https://www.zotero.org/support/quick_start_guide lub któryś z wielu dostępnych w sieci tutoriali.

⁶Zwany też systemem „autor-rok”, w którym cytowanie wyglądałoby następująco „(Lore, 2007)”.

`\printbibliography`

Komenda ta automatycznie stworzy nowy, odpowiednio podpisany rozdział „Bibliografia” i umieści wykorzystane w pracy cytowania. Początkowo lista będzie pusta, niezależnie od liczby umieszczonych w bazie cytowań. Należy pamiętać, aby rozszerzyć proces kompilacji o odpowiednie narzędzie. `TEXStudio` automatycznie dba o uruchomienie narzędzia do przetwarzania bibliografii, należy jednak wskazać właściwe — robimy to w `TEXStudio` w „Opcje → Konfiguruj `TEXStudio`... → Zbuduj” i wybraniu programu Biber jako domyślne narzędzie bibliografii. Dzięki temu przy kompilacji w edytorze cytowania powinny działać automatycznie.

Jedyne, co pozostało, to użycie cytowania. Cytowania umieszczamy w tekście, tuż za zdaniem, które wymaga podania źródła, przed interpunkcją, ale w przeciwieństwie do innych przypisów, cytowanie oddzielamy od tekstu niełamliwą spacją. Napiszemy

Pewne stwierdzenie[~]`\cite{Lore2007}`.

Źródło powinno być dodane w takim miejscu i w takiej formie, aby bez niego czytane zdanie wciąż miało sens. Oznacza to, że nie należy pisać

W `\cite{Bad2000}` napisano, że tak się nie cytuje!

Po usunięciu cytowania, zdanie brzmi „W napisano, że tak się nie cytuje!”, co nie jest prawidłowe. Lepszą opcją byłoby

Cytując zwracamy uwagę na to, jak zdanie się czyta[~]`\cite{Bad2000}`.

Pamiętajmy, że cytowanie jest odniesieniem do materiału pomocniczego, a nie częścią treści.

W pracach dyplomowych, naukowych i raportach, należy podawać źródła przy:

- definicjach, twierdzeniach, lematach i innych faktach, które przytaczamy w dokładnych słowach,
- opiniach, teoriach, predykcjach, ocenach, które nie są wnioskami do własnej analizy lub syntezy, niezależnie od tego, czy przytaczamy dokładne słowa czy parafrazujemy,
- opisach studium przypadku, statystykach i danych liczbowych, których sami nie zebraliśmy, ale na które się powołujemy,
- ilustracjach, które wykorzystujemy,
- rezultatach cudzej pracy, na której bazujemy lub którą wykorzystujemy,
- przeglądzie literatury i przy odsyłaniu do literatury uzupełniającej,
- innych miejscach, które uznajemy za stosowne.

Dobre źródło to takie, w którym jesteśmy w stanie określić autora, miejsce i czas, w którym dany materiał źródłowy został opublikowany.

Szczególną uwagę musimy poświęcić kwestiom załączania ilustracji lub rysunków. Zdjęcie bez zgody twórcy możemy wykorzystać na użytek prywatny, możemy również korzystać z tak zwanego wyjątku edukacyjnego lub dozwolonego użytku szkolnego. Zwalnia to, zrazem pracowników, jak i uczniów czy studentów instytucji prowadzących działalność dydaktyczną oraz naukową, z konieczności uzyskania zgody autora na wykorzystanie utworu. Aby skorzystać z wyjątku edukacyjnego, musi być spełnionych między innymi kilka warunków:

- utwór musi być rozpowszechniony, czyli udostępniony przez autora za jego zgodą — utwory udostępnione bez zgody autora nie podlegają pod wyjątek edukacyjny;
- celem wykorzystania utworu jest ilustracja treści przekazywanych w procesie dydaktycznym lub prowadzenie badań naukowych — na przykład wolno wykorzystać zdjęcie osoby przy notce biograficznej o niej, ale wykorzystanie zdjęcia jako przerywnika na slajdzie, nie podlega pod wyjątek edukacyjny, ponieważ pełni ono wtedy rolę estetyczną, a nie jest ilustracją przekazywanych treści;
- jeśli wykonujemy reprodukcję dowolnymi metodami w tym cyfrowymi, możemy wykorzystać drobny utwór lub jedynie fragment większego utworu — niestety ustawa nie definiuje precyzyjnie, kiedy utwór jest „drobny” i jak duży fragment podlega pod wyjątek, należy w tej materii poruszać się ostrożnie, ponieważ istnieje wiele interpretacji — w razie wątpliwości, najlepiej zapytać kogoś z większym doświadczeniem w wyborze źródeł;
- wykorzystanie utworu nie może utrudniać korzystania z niego, ani przynosić uszczerbku interesom autora — punkt ten to dwie trzecie tak zwanego testu trójstopniowego, który pomaga określić, czy zachodzi wyjątek dozwolonego użytku;
- musimy uhonorować autorskie prawa osobiste autora, podając w źródle dane autora oraz tytuł utworu — wyjątek nie zwalnia nas z obowiązku podawania źródeł.

Jak widać, musi zachodzić szereg warunków, w szczególności powód wykorzystania źródła musi być związany z dydaktyką lub nauką, musimy mieć pewność, że autor rozpowszechnił utwór intencjonalnie, musimy zacytować źródło z wymienieniem autora i w końcu, wykorzystanie źródła nie może przynieść szkody autorowi utworu.

Jeśli któryś z punktów o dozwolonym użytku edukacyjnym nie jest spełniony, mamy możliwość skontaktowania się z autorem, ponieważ jeśli uzyskamy zgodę autora, wolno nam wykorzystać utwór w zakresie wyszczególnionym w takiej zgodzie. Jeśli nie ma możliwości skontaktowania się autorem, ponieważ na przykład nie żyje, musimy liczyć na to, że utwór jest w tak zwanej domenie publicznej, w której korzystanie z utworów jest nieograniczone, przy zachowaniu praw osobistych autora, czyli podaniu autorstwa i tytułu utworu. W przypadku utworów anonimowych, powstałych w ramach stosunku pracy lub gdy utwór nie został opublikowany za życia twórcy, przechodzi do domeny publicznej po 70 latach od publikacji. Jeśli wiadomo, że wszyscy autorzy utworu zmarli dawniej niż 70 lat temu, utwór również może być traktowany jako utwór w domenie publicznej. Gdy nie wiadomo kiedy zmarł autor, utwór przechodzi do domeny publicznej po 140 latach od jego

publikacji. Od niedawna promowane jest przenoszenie utworów do domeny publicznej za życia twórców za ich zgodą — odbywa się to przez udzielenie odpowiedniej licencji. Najswobodniejszą licencją jest Creative Commons Zero, w której autor zrzeka się wszystkich praw, w szczególności również pozwala na wykorzystanie utworu bez podania źródła.

Oprócz Creative Commons Zero istnieje szereg innych licencji Creative Commons, które pozwalają na wykorzystanie utworu w pracy pod warunkiem uznania autorstwa. Wszystkie licencje Creative Commons wypisane poniżej wymagają podawania autora. Z punktu widzenia wykorzystania w pracach naukowych i dydaktyce, jeśli nie zachodzi wyjątek edukacyjny, można wykorzystać utwory na licencjach:

1. CC-BY, najbardziej liberalnej licencji z uznaniem autorstwa,
2. CC-BY-NC, wariantu CC-BY z zastrzeżeniem, że zgoda na wykorzystanie jest tylko do celów niekomercyjnych,
3. CC-BY-ND, wariantu CC-BY z zastrzeżeniem, że nie wolno modyfikować utworu — przy czym między innymi kadrowanie, odbicie, zestawienie z innymi utworami w kolaż lub mozaikę, stanowią już utwór zależny, na co ten rodzaj licencji nie pozwala,
4. CC-BY-NC-ND, wariantu CC-BY-ND z zastrzeżeniem, że zgoda na wykorzystanie jest tylko do celów niekomercyjnych.

Powinniśmy starać się unikać licencji CC-BY-SA i CC-BY-NC-SA, ponieważ licencje te narzucają na nas konieczność opublikowania utworu zależnego na tej samej licencji, co wykorzystane utwór pierwotny. Niestety, nie zawsze jest to możliwe, jeśli zamierzamy utwór opublikować, ponieważ wydawca może mieć określone zasady dotyczące możliwych licencji publikowanego utworu⁷. Na stronie <https://search.creativecommons.org/> mamy możliwość wyszukiwania materiałów w Internecie, które są opublikowane na konkretnej licencji. Zdecydowanie ułatwia to ocenę, które źródła można legalnie wykorzystać.

Wyszukiwarki takie, jak <https://search.creativecommons.org/> ułatwiają odnalezienie zdjęć, filmów i utworów muzycznych, które możemy wykorzystać. Na szczęście istnieją również bazy artykułów z podobnymi wyszukiwarkami. Politechnika Wrocławska licencjonuje wiele takich baz⁸, przy czym wiele z nich jest dostępnych za pośrednictwem HAN⁹. Do najważniejszych dla studentów i pracowników wydziału matematyki baz, można zaliczyć MathSciNet <https://mathscinet.ams.org/> oraz zbMATH <https://zbmath.org/>. Bazy te posiadają miliony prac naukowych z matematyki, wiele z nich opatrzonych licencjami oraz danymi bibliograficznymi. Bazy te również pozwalają na przeglądanie profili autorów i czasopism. Nie wszystkie prace są dostępne, ale jeśli jakaś praca ma wersję elektroniczną, to znajdzie się tam również link. Oba portale są elektronicznymi wersjami papierowych czasopism publikujących recenzje artykułów z zakresu matematyki. Na szczególną uwagę od tego roku zasługuje zbMATH, który promuje otwarty dostęp i w przeciwieństwie

⁷Więcej na temat licencji Creative Commons można przeczytać na stronie <https://creativecommons.pl/>.

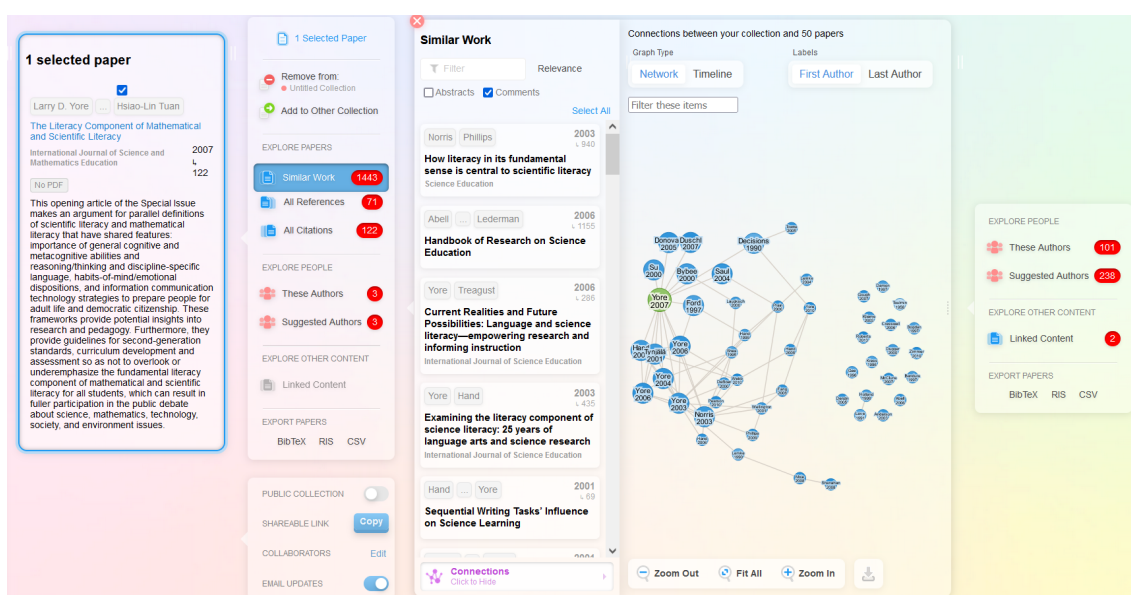
⁸Lista baz dostępna jest na <https://biblioteka.pwr.edu.pl/e-zasoby/bazy-danych>.

⁹Informacje jak uzyskać dostęp przez HAN są na stronie <https://biblioteka.pwr.edu.pl/e-zasoby/zdalny-dostep---han>.

do MathSciNet, jest otwarty dla wszystkich. Posiada również interesującą funkcjonalność wyszukiwania artykułów po formułach zapisanych w \LaTeX u oraz wyszukiwania po wykorzystanych pakietach matematycznych lub językach programowania. Oba portale korzystają z klasyfikacji tematów z zakresu matematyki o nazwie MSC 2020 (ang. Mathematics Subject Classification), które różnym dziedzinom przyporządkowuje numery¹⁰. Znając numer w klasyfikacji, możemy w tych bazach wyświetlić wszystkie artykuły z danego tematu. Przykładowo, możemy sprawdzić, że 97C60 odpowiada za socjologiczne aspekty uczenia w odniesieniu do dydaktyki matematyki¹¹, co pozwala na dotarcie do 54 publikacji na ten temat. Wszystkie artykuły w zbMATH i MathSciNet udostępniają gotowe do pobrania pliki .bib z informacjami bibliograficznymi w formacie \LaTeX a.

Jeśli interesuje nas szerszy przegląd literatury, nie tylko w ramach matematyki, możemy skorzystać z innej bazy wypisanych na stronie biblioteki Politechniki Wrocławskiej. Można też skorzystać z wyszukiwarki Google Scholar¹². Wyszukiwarka ta przeszukuje Internet w poszukiwaniu artykułów naukowych, pozwala na sprawdzanie linków do artykułów cytowanych przez i cytujących prace oraz udostępnia dane bibliograficzne w formacie Bib \TeX . Należy uważać, ponieważ baza ta tworzona jest automatycznie i niekiedy zdarzają się błędy przetwarzania, szczególnie w przypadku starszych artykułów, których skany trudniej przetwarzać automatycznie.

Ponieważ artykuły naukowe cytują źródła i same są cytowane, można myśleć o nich jak o grafie. Jeśli znamy dane artykułu, na przykład jego DOI, które możemy znaleźć w zbMATH lub MathSciNet, korzystając z ResearchRabbit¹³, możemy zwizualizować taki graf.



¹⁰Ich listę można sprawdzić pod adresem <https://mathscinet.ams.org/msnhtml/msc2020.pdf> lub <https://zbmath.org/classification/>.

¹¹<https://zbmath.org/classification/?q=cc%3A97C60>.

¹²Google Scholar dostępny jest pod adresem <https://scholar.google.com/>.

¹³ResearchRabbit dostępny jest pod adresem <https://www.researchrabbit.ai/>.

Takie narzędzia pozwalają szybko znaleźć artykuły powiązane ze sobą tematycznie i często są pierwszym krokiem, gdy uczymy się nowych narzędzi lub opracowujemy bibliografię do pracy dyplomowej, na podstawie kilku początkowych źródeł, które otrzymujemy od promotora.

Jeśli ustalimy, które artykuły chcemy zdobyć, możemy zależnie od szczęścia uzyskać do nich dostęp lub nie. Jeśli artykuł będzie dostępny w otwartych bazach, takich jak Project Euclid (<https://projecteuclid.org/>), najpewniej uzyskamy do niego dostęp bez trudu. Jeśli będzie w bazach zamkniętych, takich jak SpringerLink (<https://link.springer.com/>), JSTOR (<https://www.jstor.org/>) lub Science Direct (<https://www.sciencedirect.com/>), pomocny będzie dostęp przez HAN. Jeśli nie uda się znaleźć artykułu, możemy szukać w bazie arXiv (<https://arxiv.org/>), jednak musimy uważać, ponieważ baza ta zawiera preprinty, czyli artykuły, które nie ukazały się jeszcze w publikacji w ostatecznej formie i często są przed recenzją. W ostateczności możemy założyć konto na ResearchGate (<https://www.researchgate.net/>) i sprawdzić, czy ktoś dany artykuł udostępnił lub poprosić o jego udostępnienie. Zdarza się, że autor odpowie na taką prośbę i udostępni kopię artykułu w pliku PDF.

2 Rysunki i tabele

Rysunki i tabele z punktu widzenia składu są bardzo podobnymi typami materiałów. Jeśli wyobrazimy sobie obrys tabeli, można myśleć o prostokątnym obiekcie na kartce papieru, zbliżonym formą i przeznaczeniem do rysunku.

Pierwszym krokiem w ustalaniu sposobu umieszczenia ilustracji w tekście, jest zastanowienie się nad ich charakterem. Wyróżniamy trzy rodzaje ilustracji i tabel:

1. materiały ściśle związane z tekstem głównym, konieczne do jego zrozumienia lub stanowiące fragment tekstu głównego,
2. materiały pomocnicze, luźno związane z tekstem głównym, pomagające zrozumieć tekst główny,
3. materiały zdobiące, swoim charakterem i stylem nawiązujące do treści epoką, nastrojem i temu podobnymi.

Materiały ściśle lub luźno powiązane z tekstem nazywamy ilustracjami. W publikacjach technicznych i naukowych, ilustracje graficzne i tabelaryczne powinny być podpisane, najczęściej zaraz pod ilustracją. Powinny być również numerowane. Materiały zdobiące zwykle występują bez podpisów. Materiały ściśle związane z tekstem powinny być dokładnie w miejscu, które dyktuje treść. Materiały pomocnicze, aby nie burzyć rytmu treści, powinny być oddalone, choć wciąż możliwie blisko miejsca, które dyktuje treść. Przez możliwie blisko mamy na myśli na tej samej stronie lub którejś z następnych. Rozmieszczenie materiałów zdobiących jest poddyktowane kompozycją strony — jednakże w pracach dyplomowych,

naukowych i raportach z zakresu matematyki, materiały zdobiące zwykle nie występują, nie będziemy się więc nimi zajmować. Warto zwrócić uwagę, że większość ilustracji umieszczanych w tekstach naukowych nie jest silnie powiązane z treścią, a jedynie stanowi materiał pomocniczy.

Na przykład, jeśli prowadzimy wywód matematyczny i definiujemy pewną funkcję, jej wykres nie jest częścią treści, a jedynie ilustracją definicji. Gdy opisujemy, gdzie znajdują się ekstrema tej funkcji, podamy argumenty, w których one występują i możemy odwołać się do ilustracji, informując czytelnika, że można je zobaczyć na ilustracji o danym numerze i na danej stronie. Za każdym razem, należy się zastanowić, czy tekst bez ilustracji jest wciąż zrozumiały i wartościowy. Jeśli tak, ilustracja zapewne jest tylko materiałem pomocniczym.

Ilustracje różnią się również sposobem umieszczenia. Ilustracja może być włączona w tekst lub obłamana tekstem. Ilustracje włączone w tekst przerywają naturalny bieg tekstu podobnie jak wyróżnione wzory. Ilustracje obłamane tekstem są otoczone tekstem na lewo, na prawo lub po obu stronach. Ilustracja obłamana może występować wewnątrz tekstu, może występować u góry strony, na dole strony lub na osobnej stronie przeznaczonej na ilustracje — tak zwanej wkładce, często drukowanej osobno, być może na innej jakości papierze lub w kolorze, podczas gdy reszta publikacji jest drukowana w odcieniach szarości z powodów ekonomicznych.

Ilustracje luźno powiązane tekstem mogą być:

1. włączone w tekst u góry strony,
2. włączone w tekst u dołu strony,
3. na osobnej wkładce na ilustracje.

Ilustracje ściśle powiązane z tekstem mogą być:

1. obłamane tekstem w miejscu występowania,
2. włączone w tekst w miejscu występowania.

Rozmieszczeniem ilustracji luźno powiązanych \LaTeX zajmuje się automatycznie z wykorzystaniem tak zwanych elementów pływających. Stara się on dobrać najlepsze możliwe miejsce na ilustrację.

Aby stworzyć ilustrację w elemencie pływającym, wykorzystujemy

```
\begin{figure}[umieszczenie]
... % obrazek
\caption[tytuł do spisu]{Podpis}
\label{etykieta}
\end{figure}
```

dla obrazków oraz

```
\begin{table}[umieszczenie]
... % tabela
```

```

\caption[tytuł do spisu]{Podpis}
\label{etykietka}
\end{table}

```

dla tabel. Opcjonalny argument „umieszczenie” informuje \LaTeX a, jakie są nasze preferencje dotyczące umieszczenia ilustracji. Jest to ciąg liter, które oznaczają

t od „top”, czyli u góry strony,

b od „bottom”, czyli u dołu strony,

p od „page”, czyli na osobnej stronie.

Domyślna wartość umieszczenia to `tbp`, czyli \LaTeX najpierw spróbuje umieścić materiał pomocniczy u góry strony, jeśli to się nie uda, na dole, a jeśli to się nie uda, na osobnej stronie. Komenda `caption` dodaje podpis pod rysunkiem lub tabelą, odpowiednio je numerując. Tytuł do spisu możemy pominąć, wtedy jest taki sam jak podpis. Etykietka pozwala nam odwołać się do ilustracji, tak jak robiliśmy to dla rozdziałów i sekcji oraz wzorów.

Rozmieszczenie elementów pływających korzysta z następujących zasad:

1. ilustracja zostanie umieszczona tak blisko początku dokumentu, jak to możliwe
2. ilustracja zostanie umieszczona na stronie nie wcześniejszej, niż miejsce jej definicji,
3. ilustracja nie zostanie umieszczona przed późniejszymi tego samego typu, czyli w szczególności wszystkie rysunki będą występować w kolejności definiowania, wszystkie tabele również w kolejności definiowania, ale pomiędzy nimi kolejność nie musi być zachowana,
4. ilustracja może być umieszczona w miejscu określonym przez umieszczenie,
5. umieszczenie ilustracji nie może spowodować przepełnienia strony,
6. u góry strony wolno umieścić maksymalnie dwie ilustracje, u dołu jedną, w sumie trzy,
7. w przypadku stron włączanych w tekst, ilustracje u góry mogą zająć co najwyżej 70% strony, u dołu 30%, tekst musi zająć minimum 20% wysokości strony, w przypadku stron na wkładce, ilustracje muszą zajmować co najmniej 50% wysokości.

Aby móc umieszczać ilustracje włączane w tekst dokładnie w miejscu definicji, czyli dla ilustracji ściśle związanych z testem, musimy załadować pakiet `float`

```
\usepackage{float}
```

Dodaje on między innymi opcję `H` do umieszczenia rysunków i tabel, oznaczającą „here”, czyli tutaj. Drugi sposób na umieszczenie ilustracji ściśle związanych z tekstem to ilustracje obłamane tekstem. Do ich umieszczenia ładuje się pakiet `wrapfig`

```
\usepackage{wrapfig}
```

Po załadowaniu pakietu oprócz `figure` i `table` mamy dwa środowiska — `wrapfigure` oraz `wraptable`.

```
\begin{wrapfigure}{pozycja}{szerokość}
... % obrazek
\caption[tytuł do spisu]{Podpis}
\label{etykieta}
\end{wrapfigure}
```

Środowisko dla tabel definiuje się analogicznie. W powyższym przykładzie pozycja to jedna z liter `r`, `l`, `i` oraz `o`, które oznaczają odpowiednio prawa, lewa, wewnętrzna oraz zewnętrzna strona kolumny. Wewnętrzna i zewnętrzna funkcjonuje tylko w dokumentach, które są dwustronne, czyli mają niesymetryczne marginesy — są to na przykład `book` i `mwbk`. Szerokość ilustracji możemy podać za pomocą miar odległości takich jak `6cm` lub relatywnie do szerokości kolumny, na przykład `0.3\textwidth` spowoduje, że ilustracja będzie miała 30% szerokości kolumny tekstu.

Pakiety `float` i `wrapfig` mają więcej opcji, do zapoznania się z którymi zachęcam. Większość potrzebnych informacji możemy znaleźć na stronach dokumentacji pakietów¹⁴.

Zwróćmy uwagę, że środowiska do umieszczania ilustracji ściśle związanych z tekstem nie są dostępne domyślnie, tylko musimy załadować je odpowiednimi pakietami. Dzieje się tak, ponieważ co już było podkreślane, większość ilustracji spotykanych w dokumentach technicznych to w rzeczywistości materiały pomocnicze, a nie treść sama w sobie, zatem powinniśmy umieszczać je w elementach pływających.

Pomocny przy sterowaniu elementami pływającymi jest też pakiet `placeins`.

```
\usepackage{placeins}
```

Pakiet ten dostarcza komendy `\FloatBarrier`, której użycie powoduje, że wymuszamy pojawienie się wszystkich ilustracji dotąd zdefiniowanych przed wystąpieniem komendy `\FloatBarrier`. Jest to bariera, której elementy pływające nie mogą przekroczyć. Opcje pakietu pozwalają na automatyczne dodanie tej komendy do wszystkich sekcji, dzięki czemu mamy pewność, że ilustracje definiowane w danej sekcji pokażą się w tekście przed jej końcem.

```
\usepackage[section]{placeins}
```

Teraz gdy wiemy już jak sterować umieszczeniem ilustracji, podpisami oraz etykietkami, pozostaje nauczyć się jak tworzyć obrazki i tabele. Do załączania obrazków polecany jest pakiet

```
\usepackage{graphicx}
```

Dostarcza on komendę

```
\includegraphics[opcje]{nazwa_pliku}
```

¹⁴Dokumentacja dostępna jest na stronie <https://ctan.org/pkg/float> oraz <https://ctan.org/pkg/wrapfig>.

gdzie opcje to na przykład `width=6cm`, jeśli chcemy, aby obrazek miał szerokość 6 centymetrów. Nazwa pliku powinna być relatywna, zaczynająca się od katalogu, w którym znajduje się plik `.tex`. Do załączania zalecane są pliki `.pdf`, a jeśli nie jest to możliwe, `.png`. Należy unikać plików `.jpeg`, ponieważ w większości z nich wykorzystana jest stratna kompresja, która zmniejsza rozmiar pliku kosztem jego jakości. Najlepsze efekty uzyskamy tworząc grafiki w formacie `.pdf` w programach do grafiki wektorowej, lub zapisując na przykład wykresy w formacie wektorowym do `.pdf`.

Istnieje również możliwość tworzenia ilustracji w sposób programistyczny. Służy do tego pakiet TikZ. Zachęcam do zapoznania się z galerią przykładów TikZ na stronie internetowej <https://texample.net/tikz/examples/>. Galeria przykładów TikZ posiada przykładowe kody, które możemy skopiować, uruchomić, zmodyfikować i wykorzystać do nauki korzystania z pakietu. Przydatnym materiałem do nauki TikZa jest również „A very minimal introduction to TikZ” autorstwa J. Créméra, dostępny na stronie internetowej autora pod adresem <https://cremeronline.com/LaTeX/minimaltikz.pdf>.

Tabele tworzymy za pomocą środowiska `tabular`. Kolumny oddzielamy za pomocą znaku `&`, wiersze za pomocą `\\`. Dodatkowo przydatne mogą być `\hline` (tworzące linię poziomą) oraz `\cline{od-do}` (tworzące linię poziomą we wskazanych kolumnach). Liczbę kolumn oraz wyrównywanie deklarujemy przez opcję do środowiska. Wpisujemy tam ciąg znaków, oznaczający kolejno:

- `|` (linia pionowa),
- `l` (kolumna wyrównywana do lewej),
- `r` (kolumna wyrównywana do prawej),
- `c` (kolumna wyśrodkowana),
- `p{6cm}` (kolumna konkretnej szerokości — tu 6cm) oraz
- `@{tekst}` (znak oddzielający kolumny).

Istnieje możliwość scalenia komórek w jedną. Aby scalić komórki w poziomie, używamy komendy `\multicolumn{n}{a}{c}`, gdzie `n` jest liczbą kolumn do scalenia, a nową specyfikacją wyrównywania dla scalonej komórki, natomiast `c` jest zawartością scalonej komórki. Jeśli do dokumentu załączymy pakiet `multirow`, będziemy również mogli łączyć komórki w pionie.

Poniżej kilka przykładów tabel, tutaj bez środowiska `table` lub `wraptable`.

```
\begin{tabular}{c|c|c}
  \textbf{Nagłówek 1} & & \\
  \textbf{Nagłówek 2} & & \\
  \textbf{Nagłówek 3} & \\\ \hline
  1 & 2 & 3 \\\
  test & Hello & World
\end{tabular}
```

Nagłówek 1	Nagłówek 2	Nagłówek 3
1 test	2 Hello	3 World

Kolejny przykład demonstruje zastosowanie podwójnych linii oraz kolumny tekstu na zadaną szerokość.

```
\begin{tabular}{|r|p{5cm}|}\hline
  Kto & powiedział \\ \hline\hline
  Hans Freudental & Ze wszystkich nauk humanistycznych
  matematyka jest najbardziej humanistyczna. \\
  Fox & Jacha-chacha-chacha-chow! \\ \hline
\end{tabular}
```

Kto	powiedział
Hans Freudental	Ze wszystkich nauk humanistycznych matematyka jest najbardziej humanistyczna.
Fox	Jacha-chacha-chacha-chow!

W końcu przykład bardziej zaawansowanej tabeli, z łączeniem kolumn oraz wyrównaniem na przecinku.

```
\begin{tabular}{l|r@{,}l}
  \multicolumn{1}{c|}{\textbf{Produkt}} & & \\
  \multicolumn{2}{c}{\textbf{Cena}} \\ \hline
  Rzecz A & 9 & 99 \\
  Inna rzecz & 1001 & 04 \\
  Jeszcze inna rzecz & 0 & 07
\end{tabular}
```

Produkt	Cena
Rzecz A	9,99
Inna rzecz	1001,04
Jeszcze inna rzecz	0,07

Więcej przydatnych informacji na temat tabel w \LaTeX u, możemy znaleźć w Wiki-Książce poświęconej \LaTeX owi¹⁵.

3 Kody programów

Aby umieścić kody programów, możemy albo użyć środowiska `lstlisting` pakietu `listings`.

¹⁵Rozdział poświęcony tabelom dostępny jest na stronie <https://en.wikibooks.org/wiki/LaTeX/Tables>.

```
\usepackage{listings}
```

Pakiet listings posiada zdefiniowane wyróżnianie składni dla wielu języków, przykładowo dla Pythona.

```
\begin{lstlisting}[language=Python]
def funkcja(x=0):
    """Dokumentacja funkcji"""
    return lambda y: x+y # komentarz
\end{lstlisting}
```

Da w wyniku

```
def funkcja(x=0):
    """Dokumentacja funkcji"""
    return lambda y: x+y # komentarz
```

Zwróćmy uwagę, że słowa kluczowe są pogrubione, a komentarze są zapisane krojem pochyłym.

Kody programów zwykle stanowią materiał ściśle związany z treścią, dlatego domyślnie pojawiają włamane w tekst w miejscu występowania. Możemy umieścić je w środowisku pływającym, dodając opcję float przy początku środowiska.

```
\begin{lstlisting}[language=Python,float]
...
\end{lstlisting}
```

Wtedy oczywiście konieczne jest też dodanie podpisu oraz najlepiej etykiety, aby odwołać się do konkretnego programu. Aby dodać podpis na dole i stworzyć etykietkę, napiszemy

```
\begin{lstlisting}[language=Python,float,caption={...},label={...},captionpos=b]
...
\end{lstlisting}
```

Do wyróżniania komend w tym samym stylu, ale wewnątrz tekstu, służy komenda `\lstinline`.

Pakiet listings ma bardzo dużo opcji, warto więc zapoznać się z jego dokumentacją, tym bardziej że wiele z nich można ustawić jako domyślne dla całego dokumentu¹⁶.

W przypadku prostych kodów, pakiet listings może być zbyt dużą armatą do wróbla. W takim przypadku możemy skorzystać ze środowiska `verbatim`.

```
\begin{verbatim}
...
\end{verbatim}
```

Przy tym środowisku wszystko, co zostanie umieszczone wewnątrz niego, zostanie wpisane wprost, bez „uruchamiania” komend \LaTeX a. Wariantem wewnątrz linii, do umieszczenia komend wprost bez uruchamiania, jest komenda `\verb`. Działa ona nieco inaczej, niż pozostałe, ponieważ sprawdza, jaki znak jest następny po komendzie i jako argument,

¹⁶Dokumentacja dostępna jest na stronie pakietu <https://www.ctan.org/pkg/listings>.

traktuje wszystko do kolejnego wystąpienia tego samego znaku. Przykładowo prawidłowe są `\verb|...|` oraz `\verb+...+`. Dzięki temu możemy wybrać taki znak, jaki nie występuje w komendzie. W szczególności, aby złożyć `\verb|...|`, w kodzie \LaTeX a wykorzystane zostało `\verb+\verb|...|+`.

Alternatywą dla wymienionych pakietów jest pakiet `minted`, jednakże wymaga on zainstalowanego w systemie Pythona — musi być widoczny w ścieżce — oraz biblioteki `Pygments`¹⁷, która rozpoznaje składnie wielu języków i na tej podstawie odpowiednio koloruje wyrażenia. Rozwiązanie to nie jest dostępne we wszystkich instalacjach \LaTeX a, dlatego nie jest rekomendowane na wykładzie jako domyślne. Jeśli jednak posiadamy zainstalowanego Pythona z biblioteką `Pygments`, to pakiet ten prawdopodobnie pozwoli uzyskać ładniejsze formatowanie kodu. Przykładowo pisząc

```
\begin{minted}{Python}
import os

os.chdir(os.path.join(os.path.expanduser("~/"), "Downloads"))

suma = 0
najlepsza = 0

with open("zadanie1.txt") as f:
    while linia := f.readline():
        linia = linia.strip()
        if linia:
            suma += int(linia)
        else:
            najlepsza = max(najlepsza, suma)
            suma = 0

wynik = max(najlepsza, suma)
print("Odpowiedź to", wynik)
\end{minted}
```

uzyskamy złożony program wyglądający jak poniżej.

```
import os

os.chdir(os.path.join(os.path.expanduser("~/"), "Downloads"))

suma = 0
najlepsza = 0

with open("zadanie1.txt") as f:
```

¹⁷Patrzy <https://pygments.org/> oraz <https://pypi.org/project/Pygments/>.


```

while linia := f.readline():
    linia = linia.strip()
    if linia:
        suma += int(linia)
    else:
        najlepsza = max(najlepsza, suma)
        suma = 0

wynik = max(najlepsza, suma)
print("Odpowiedź to", wynik)

```

Warto zwrócić uwagę, że domyślnie `minted` korzysta z różnych kolorów, co może utrudniać proces drukowania i zwiększać jego koszt w porównaniu do druku czarno-białego. Wyłączenie kolorów jest możliwe, ale nie jest proste, ponieważ kolory są kodowane jako wartości liczbowe przez bibliotekę `Pygments` i po stronie systemu \LaTeX nie są zmiennymi, którymi da się łatwo sterować¹⁸.

4 Prezentacje

Temat prezentacji nie będzie poruszony podczas zajęć, jednakże zachęcam, by zgłębić tę tematykę samodzielnie. W niniejszej sekcji kilka informacji, które mogą być pomocne.

Do tworzenia prezentacji w \LaTeX u służy klasa dokumentów `beamer`. Posługujemy się nią niemal tak samo, jak każdą inną klasą, z tą różnicą, że sami definiujemy slajdy za pomocą środowiska `frame`, a nie pozwalamy dzielić materiał \LaTeX owi na strony. Sporo informacji o robieniu prezentacji znajdziecie w Wiki-Książce o \LaTeX u w rozdziale poświęconym prezentacjom, pod adresem <https://en.wikibooks.org/wiki/LaTeX/Presentations>.

Osoby szukające stylu zgodnego z logotypem Politechniki Wrocławskiej, zachęcam do zapoznania się ze stylem Wojciecha Myszk¹⁹ lub moim, nieco lżejszym stylem i dalszym od oficjalnej książki logotypu, ale odrobinę nowocześniejszym wizualnie²⁰.

5 Integracja Python – \LaTeX

5.1 Sprawdzenie środowiska

Jeśli zgodnie z instrukcją zainstalowaliście Anacondę oraz `TeXLive` i korzystamy z Windowsa, wszystko powinno być gotowe. Możemy to sprawdzić, uruchamiając w menu start standardowy terminal Windowsa (`cmd.exe`) oraz wpisując

¹⁸Osoby, które potrzebują ustawić styl czarno-biały, mogą skorzystać z porady z Internetu <https://tex.stackexchange.com/questions/548592/changing-all-colors-to-black-white-using-minted-sty> oraz dokumentacji <http://tug.ctan.org/macros/latex/contrib/minted/minted.pdf>.

¹⁹<https://kmim.wm.pwr.edu.pl/myszka/projekty/szablon-prezentacji-zgodny-z-ksiega-logotypu-politechniki-wroclawskiej/>.

²⁰<https://github.com/aginiewicz/beamer-pwrlite>.

```
"C:\Users\[Nazwa Użytkownika]\Anaconda3\Scripts\Activate.bat" && pythontex --version
```

gdzie [Nazwa Użytkownika] to nazwa użytkownika korzystającego z komputera, na przykład admin.

Jeśli na ekranie pojawi się informacja, że PythonTeX jest w wersji przynajmniej 0.18, to komenda do wpisania w konfigurację edytora będzie następująca:

```
"C:\Users\[Nazwa Użytkownika]\Anaconda3\Scripts\Activate.bat" && pythontex %.tex
```

Na innych systemach operacyjnych wszystkie komendy zwykle są w ścieżce, wtedy

```
pythontex --version
```

powinno zadziałać bez niczego z przodu i komenda do wpisania w konfigurację edytora a będzie następująca:

```
pythontex %.tex
```

5.1.1 Co gdy to powyżej nie zadziałało?

W innych konfiguracjach (jeśli ktoś nie podążył za instrukcją instalacji), podczas budowy komendy musimy zadbać o dwie rzeczy:

1. mieć dostęp do PythonTeXa,
2. mieć dostęp do Pythona.

Jeśli nie mamy dostępu do PythonTeXa w ścieżce (komenda pythontex nie jest w ścieżce²¹), musimy go znaleźć. Zapewne będzie w miejscu, w którym zainstalowaliśmy TeX a, na przykład w katalogu C:\texlive\2024\texmf-dist\scripts\pythontex\pythontex3.py lub /usr/share/texmf-dist/scripts/pythontex/pythontex3.py. Jeśli nie wiesz, gdzie znaleźć plik, poszukaj go, używając wyszukiwarki plików w systemie. Wtedy we wszystkich komendach, zamiast pythontex, musisz napisać pełną ścieżkę, czyli na przykład

```
C:\texlive\2024\texmf-dist\scripts\pythontex\pythontex3.py
```

Uwaga, jeśli w nazwie są spacje, pamiętaj wziąć ją w cudzysłów.

Jeśli nie mamy dostępu do Pythona w ścieżce (komenda python nie jest w ścieżce) mamy dwie opcje:

1. jeśli korzystamy z Anacondy, powinniśmy aktywować wirtualne środowisko,
2. jeśli nie korzystamy z Anacondy, musimy znać ścieżkę do Pythona.

Jeśli aktywujemy środowisko, komenda będzie wyglądała tak:

²¹Pamiętaj o komendzie |which|, która może pomóc w sprawdzeniu, czy coś jest w ścieżce.

aktywowanie środowiska `&& pythontex %.tex`

jeśli natomiast znamy ścieżkę do Pythona, komenda będzie wyglądała tak:

`ścieżka_do_pythona ścieżka_do_pythontex3.py %.tex`

Informacje o tym, jak aktywować środowisko znajdują się we wcześniejszych notatkach. Aby sprawdzić, czy dobrze ustaliliśmy ścieżkę, możemy uruchomić komendę podając `--version` zamiast `%.tex`.

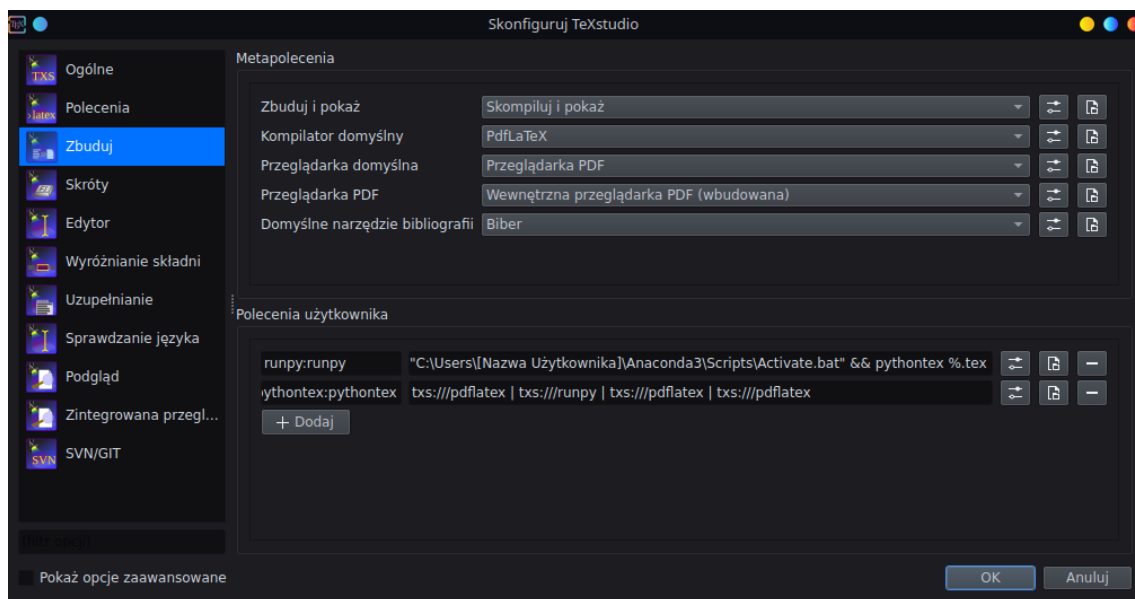
6 Konfiguracja TeXStudio

TeXStudio domyślnie nie będzie wiedział, gdzie szukać Pythona ani nawet, że chcemy używać Pythona. Możemy jednak stworzyć nowe komendy, które pomogą w przetwarzaniu dokumentów z PythonTeXem.

Wchodzimy do menu `Opcje > Konfiguruj TeXstudio...`, następnie wybieramy zakładkę `Zbuduj` i dodajemy dwa polecenia użytkownika. Pierwsze nazywamy `runpy:runpy` i jako jego treść podajemy komendę do uruchamiania PythonTeXa z poprzedniego punktu. Druga komenda to `pythontex:pythontex`, która uruchomi PDFLaTeXa oraz PythonTeXa. Jako zawartość komendy podajemy

```
txs:///pdflatex | txs:///runpy | txs:///pdflatex | txs:///pdflatex
```

co oznacza: uruchom PDFLaTeXa, potem PythonTeXa, potem jeszcze dwa razy PDFLaTeXa. Okno po konfiguracji może wyglądać następująco:



Konfigurację potwierdzamy przyciskiem OK.

6.1 Pierwszy dokument

Stwórzmy nasz pierwszy dokument w PythonTeXu.

```
% !TeX TXS-program:compile = txs:///pythontex
\documentclass{article}

\usepackage{pythontex}

\begin{document}

\begin{pycode}
print("Python says ``Hello!'")
\end{pycode}

$8 \times 256 = \py{8*256}$
\end{document}
```

Zwróćmy uwagę na pierwszą linię. To komentarz, który jest ignorowany przez L^AT_EXa, ale który T_EXStudio rozpoznaje i wie, że powinien do kompilacji dokumentu użyć komendy `pythontex:pythontex`, którą właśnie stworzyliśmy. Po zapisaniu klikamy przycisk kompilacji dokumentu²². Pojawi się komunikat z pytaniem, czy zgadzamy się, aby T_EXStudio używało nowej komendy do wszystkich dokumentów, na co powinniśmy się zgodzić. Jeśli wszystko poszło dobrze, powinniśmy zobaczyć nowy dokument a w nim dwie linie:

```
Python says "Hello!"
8 × 256 = 2048
```

6.2 Po co to wszystko?

Dzięki temu, mamy:

1. możliwość uruchamiania dowolnego kodu Pythona jako części dokumentu, w tym obliczeń w SymPy, generowania rysunków w Matplotlib oraz więcej,
2. możliwość łatwej kompilacji dokumentu z Pythonem jednym kliknięciem w T_EXStudio.

Więcej informacji o tym, jak używać PythonTeXa, znajdziemy w jego dokumentacji pod adresem <https://ctan.org/pkg/pythontex>. Szczególnie polecam „Quick start documentation” oraz „Gallery of use”.

²²Dla przypomnienia, dwie zielone strzałki.

7 Bibliografia dla chętnych — robienie notatek z matematyki „na żywo”

Tym razem w ramach bibliografii trochę materiałów on-line dotyczących szybkiego robienia notatek z matematyki (na równi lub szybciej niż mówi lub pisze wykładowca). Dotyczą one edytora Vim lub NeoVim oraz systemu \LaTeX . Jeśli ktoś nie czuje się gotowy na to, aby używać edytora Vim (lekcje o używaniu NeoVim są w ostatniej playliście w linkach), może zacząć korzystać z Vim’a poprzez edytor VS Code — <https://www.barbarianmeetscoding.com/boost-your-coding-fu-with-vscode-and-vim/> — może być to przyjaźniejsza forma na początek, ale trudniej uzyskać tak duże prędkości, żeby notować *na żywo*.

1. Elijan Mastnak — A guide to supercharged mathematical typesetting
<https://www.ejmastnak.com/tutorials/vim-latex/intro/>;
2. Gilles Castel — How I’m able to take notes in mathematics lectures using LaTeX and Vim
<https://castel.dev/post/lecture-notes-1/>;
3. Gilles Castel — How I draw figures for my mathematical lecture notes using Inkscape
<https://castel.dev/post/lecture-notes-2/>;
4. Gilles Castel — How I manage my LaTeX lecture notes
<https://castel.dev/post/lecture-notes-3/>;
5. Benjamin Brast-McKie — Demonstration of Using NeoVim for Academic Writing,
<https://www.youtube.com/playlist?list=PLBYZ1xfnKeDToZ2XXbUGSC7BkorWdyNh3l>;
6. ThePrimeagen — Vim as Your Editor
https://www.youtube.com/playlist?list=PLm323Lc7iSW_wuxqmKx_xxNtJC_hJbQ7R;
7. TJ DeVries — The Only Video You Need to Get Started with Neovim.
<https://www.youtube.com/watch?v=m8C0Cq9Uv9o>.

8 Bibliografia dla chętnych — książki

Na koniec polecam kilka książek dotyczących \LaTeX a, oprócz polecanych wcześniej materiałów internetowych. Uszeregowane są w kolejności od najbardziej „praktycznej”, do najbardziej „teoretycznej”. Pierwsza z pozycji zasługuje na szczególną uwagę, ponieważ jest formą „książki kucharskiej”, w której każdy rozdział poświęcony jest przeglądowi przydatnych pakietów do osiągnięcia danego celu. Kolejne dwie pozycje dotyczą podstaw \LaTeX a, natomiast ostatnia zawiera opis samego czystego \TeX a.

1. Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, Christine Detig, Joachim Schrod, „The \LaTeX Companion”, Addison-Wesley Professional, Reading, Massachusetts 2004, 2 wydanie²³,

²³Wkrótce dostępne będzie wydanie trzecie — <https://www.latex-project.org/news/2023/03/17/TLC3/>.

2. Helmut Kopka, Patrick W. Daly, „Guide to \LaTeX ”, Addison-Wesley Professional, Reading, Massachusetts 2003, 4 wydanie,
3. Lamport Leslie, „ \LaTeX : Podręcznik i przewodnik użytkownika”, Wydawnictwa Naukowo Techniczne WNT 2004,
4. Donald E. Knuth, „ \TeX : Przewodnik użytkownika”, Wydawnictwa Naukowo Techniczne WNT 2005.

Szukanie pomocy warto rozpocząć na stronie <https://tex.stackexchange.com/>, na której wszystkie pytania są związane z \TeX em.