# Tvorba webu v IGCMS

Toto je výchozí obsah webu psaný v HTML+. Je navržen tak, aby dal správcům webu základní přehled o procesu jeho vytvoření. V druhé části obsahu je několik ukázek typických prvků, které se při vytváření webu mohou hodit.

Před začátkem práce se systémem je potřeba získat přehled o jeho fungování. Především je potřeba připravit kvalitní strukturovaný obsah vytvářeného webu. Technologie a vzhled výsledného webu jsou z hlediska tvorby až na druhém místě.

## Jak začít s IGCMS

Proces vytvoření webu na systému IGCMS rozdělený do několika kroků. Především jde o vytvoření obsahu se strukturou následované seznámením se systémem. V další fázi je možné dokument do systému naimportovat a v posledních fázích obsah doupravit a nastavit vzhled.

1. Připravte si dobře strukturovaný koncept (dokument) o tom, čeho chcete dosáhnout. Zaměřte se na strukturu a obsah (ne jen obsah).
   * Každý koncept má pouze jeden hlavní nadpis.
   * Po každém nadpisu by mělo následovat shrnutí následující pod/kapitoly (pro účely importu je shrnutí jako vycentrovaný odstavec za nadpisem).
   * Následuje samotný obsah, který by měl být členitý – kromě odstavců také seznamy apod. Delší kapitoly je dobré členit na podkapitoly – nadpisy nižší úrovně.
   * Obsah jako celek by měl být vyvážený tak, aby neobsahoval příliš mnoho ale ani příliš málo prvků stejného typu za sebou.
2. Před používáním systému si projděte [Specifikaci IGCMS](http://partner.internetguru.cz/igcms).
   * Login: partner / igpartner
   * Nejdříve se ujistěte, že chápete *základní pojmy*.
   * Kdykoli si nevíte rady, čtěte manuál.
3. [Importujte dokumenty](http://localhost/ig?import) do IGMS.
   * Nezapomeňte, že importovaný soubor je uložený odděleně, aby se zabránilo nežádoucímu přepsání.
   * Pokud chcete použít importovaný obsah (nebo jeho část), zkopírujte jej do souboru s obsahem. V této fázi stačí soubor zkopírovat do schránky.
   * Importovaný dokument je možné k obsahu připojit přímo pomocí atributu import. Případné další změny importovaného souboru se projeví okamžitě v obsahu webu.
4. Pro vytvoření obsahu [vstupte do administrace](http://localhost/ig?admin) nebo soubory modifikujte přímo přes FTP.
   * Do výchozího dokumentu Content.html vložte obsah ve formátu HTML+. Např. ten ze schránky – zkopírovaný v předchozím kroku z importovaného dokumentu.
5. Vzhled a další nastavení
   * V rámci odkazů ve spodní části administrace můžete procházet a měnit datové a konfigurační soubory jednotlivých rozšíření dle libosti.
   * Každý vytvořený soubor je možné deaktivovat a tím obnovit výchozí hodnoty.

## Ukázky použití

Několik příkladů typického použití systému při tvorbě webu. Použití definičního seznamu s několika řádkovými prvky a ukázkou zobrazení systémové proměnné.

* **Formátování textu**
* Text odstavce (nebo jiného prvku) může být **zdůrazněný** (typicky tučně), *méně důrazný* (typicky kurzívou), může se jednat o příklad a další.
* **Používání proměnných**
* Zobrazit hodnotu globální proměnné cms-version: $cms-version
* Zobrazit hodnotu globální proměnné cms-version: **$cms-version**
* Zobrazit hodnotu globální proměnné cms-version: <noparse>$cms-version</noparse>
* Zobrazit hodnotu globální proměnné cms-version: **<noparse>$cms-version</noparse>**
* Předchozí proměnná se zobrazí pomocí kódu ‘’<em var=“cms-version”/>’’
* Předchozí proměnná se zobrazí pomocí kódu ''<em var="cms-version"/>''
* Předchozí proměnná se zobrazí pomocí kódu <code xml><em var="cms-version"/></code>
* Předchozí proměnná se zobrazí pomocí kódu <code xml><em var=“cms-version”/></code>

CODE#1

function keywords(str) {

var obj = {}, words = str.split(" ");

for (var i = 0; i < words.length; ++i) obj[words[i]] = true;

return obj;

}

function heredoc(delim) {

return function(stream, state) {

if (stream.match(delim)) state.tokenize = null;

else stream.skipToEnd();

return "string";

};

}

CODE#2

function keywords(str) {  
 var obj = {}, words = str.split(" ");  
 for (var i = 0; i < words.length; ++i) obj[words[i]] = true;  
 return obj;  
 }  
  
 function heredoc(delim) {  
 return function(stream, state) {  
 if (stream.match(delim)) state.tokenize = null;  
 else stream.skipToEnd();  
 return "string";  
 };  
 }

CODE#3

function keywords(str) {  
 var obj = {}, words = str.split(" ");  
 for (var i = 0; i < words.length; ++i) obj[words[i]] = true;  
 return obj;  
 }

function heredoc(delim) {  
 return function(stream, state) {  
 if (stream.match(delim)) state.tokenize = null;  
 else stream.skipToEnd();  
 return "string";  
 };  
 }

CODE#4

function keywords(str) {  
 var obj = {}, words = str.split(" ");  
 for (var i = 0; i < words.length; ++i) obj[words[i]] = true;  
 return obj;  
}  
  
function heredoc(delim) {  
 return function(stream, state) {  
 if (stream.match(delim)) state.tokenize = null;  
 else stream.skipToEnd();  
 return "string";  
 };  
}

That’s it